

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

УТВЕРЖДАЮ
Проректор по учебной работе

_____ Василенко В.Н.
(подпись) (Ф.И.О.)

«25» мая 2023 г.

РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ

Программирование на Java

Направление подготовки

09.03.03 Прикладная информатика

Направленность (профиль) подготовки

Цифровизация бизнес-процессов

Квалификация выпускника

бакалавр

Воронеж

1. Цели и задачи дисциплины

1. Целью освоения учебной дисциплины «Программирование на Java» является получение знаний о современном объектно-ориентированном языке программирования Java и овладение основными приемами программирования.

Дисциплина направлена на решение задач профессиональной деятельности *производственно-технологического, проектного* типа.

Программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.03 «Прикладная информатика».

2. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения образовательной программы

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИД1 _{УК-1} - Анализирует поставленную задачу и осуществляет поиск необходимой информации для ее решения
		ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений
ПКв-8	Способность проводить тестирование компонентов программного обеспечения ИС.	ИД1 _{ПКв-8} - выявление и исправление дефектов и несоответствий в архитектуре и дизайне ИС
		ИД2 _{ПКв-8} - осуществление исправлений дефектов и несоответствий в коде ИС и документации к ИС с последующим подтверждением
		ИД3 _{ПКв-8} - настройка параметров ИС и тестирование результатов настройки
ПКв-9	Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.	ИД1 _{ПКв-9} - использование прикладного инструментария для разработки программного обеспечения в соответствии с назначением объекта проектирования
		ИД2 _{ПКв-9} – демонстрация знаний основ информационного обеспечения прикладных процессов
		ИД3 _{ПКв-9} - разработка программного и информационного обеспечения ИС

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 _{УК-1} - Анализирует поставленную задачу и осуществляет поиск необходимой информации для ее решения	Знает: синтаксис языка Java;
	Умеет: анализировать код программы, написанный на языке Java;
	Владеет: средствами программирования на Java
ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений	Знает: принцип работы с каждым типом данных;
	Умеет: применять систему ввода/вывода на Java
	Владеет: приемами документирования кода программы;
ИД1 _{ПКв-8} - выявление и исправление	Знает: возможности языка и области применения Java –

дефектов и несоответствий в архитектуре и дизайне ИС	приложений;
	Умеет: создавать консольные приложения на языке Java
	Владеет: приемами работы со встроенными типами данных;
ИД2 _{ПКв-8} - осуществление исправлений дефектов и несоответствий в коде ИС и документации к ИС с последующим подтверждением	Знает: основные типы данных языка Java;
	Умеет: сопоставлять компоненты языка Java
	Владеет: средствами программирования на Java
ИД3 _{ПКв-8} - настройка параметров ИС и тестирование результатов настройки	Знает: основные пакеты, классы, методы языка Java
	Умеет: создавать консольные приложения на языке Java
	Владеет: стандартными методами работы с файлами и каталогами
ИД1 _{ПКв-9} - использование прикладного инструментария для разработки программного обеспечения в соответствии с назначением объекта проектирования	Знает: основной синтаксис языка Java;
	Умеет: анализировать код программы, создавать компоненты языка Java
	Владеет: средствами программирования на Java
ИД2 _{ПКв-9} – демонстрация знаний основ информационного обеспечения прикладных процессов	Знает: принципы создания визуальных форм
	Умеет: работать с базами данных на языке Java
	Владеет: приемами создания визуальных приложений
ИД3 _{ПКв-9} - разработка программного и информационного обеспечения ИС	Знает: основные визуальные компоненты
	Умеет: создавать консольные приложения на языке Java
	Владеет: технологией работы с базами данных

3. Место дисциплины (модуля) в структуре ОП ВО /СПО

Дисциплина относится к *части, формируемой участниками образовательных отношений* Блока Б1.В.04.02. Дисциплина является обязательной к изучению.

Изучение дисциплины основано на знаниях, умениях и навыках, полученных при изучении обучающимися дисциплин «Математика», «Компьютерные технологии», «Информационные системы и технологии», «Алгоритмизация и программирование».

Дисциплина является предшествующей для *преддипломной практики*.

4. Объем дисциплины (модуля) и виды учебных занятий

Общая трудоемкость дисциплины (модуля) составляет 4 зачетных единиц.

Виды учебной работы	Всего часов	Семестр
		4
	акад.	акад.
Общая трудоемкость дисциплины (модуля)	144	144
Контактная работа, в т.ч. аудиторные занятия:	57,1	57,1
Лекции	18	18
<i>в том числе в форме практической подготовки</i>	-	-
Практические занятия (ПЗ)	36	36
<i>в том числе в форме практической подготовки</i>	36	36
Семинары (С)	-	-
Лабораторные работы (ЛБ)	-	-
Консультации текущие	3,1	3,1
Проведение консультаций перед экзаменом		
Виды аттестации (зачет, экзамен)		экзамен
Самостоятельная работа:	53,1	53,1

Проработка материалов по конспекту лекций	30	30
Проработка материалов по учебникам, учебным пособиям	20	20
Курсовой проект (работа)		
Реферат		
и (или) другие виды самостоятельной работы	3,1	3,1
Подготовка к экзамену (контроль)	33,8	33,8

5 Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

5.1 Содержание разделов дисциплины (модуля)

№ п/п	Наименование раздела дисциплины	Содержание раздела (указываются темы и дидактические единицы)	Трудоемкость раздела, ак. час
1	Основы Java	Основы Java	8
2	ООП в Java, наследование	ООП в Java, наследование	16
3	Операции в Java	Операции в Java	19
4	Массивы в Java	Массивы в Java	19
5	Структура Java программ	Структура Java программ	17,1
6	Коллекции элементов	Коллекции элементов	14
7	Исключительные ситуации	Исключительные ситуации	14
	Консультации текущие		3,1
	Консультации перед экзаменом		
	Экзамен		33,8

5.2 Разделы дисциплины (модуля) и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции ак. ч.	Практические занятия в традиционной форме, ак. ч.	СРО, ак. час
1	Основы Java	1	2	5
2	ООП в Java, наследование	2	4	10
3	Операции в Java	3	6	10
4	Массивы в Java	3	6	10
5	Структура Java-программ	3	6	8,1
6	Коллекции элементов	3	6	5
7	Исключительные ситуации	3	6	5
	Консультации текущие		3,1	
	Консультации перед экзаменом			
	Экзамен		33,8	

5.2.1 Лекции

№ п/п	Наименование раздела дисциплины	Тематика лекционных занятий	Трудоемкость в традиционной форме, ак. час
1	2	3	4
1	Основы Java	Основы Java. Основные достоинства языка. Основные недостатки. Основные особенности. Версии Java. Апплеты. Жизненный цикл программы на Java. Структура пакета SDK.	1
2	ООП в Java, наследование	ООП в Java, наследование. Объектно-ориентированный язык	2

		программирования. 5 принципов объектно-ориентированного подхода. Реализация принципов объектно-ориентированного подхода в Java.	
3	Операции в Java	Операции в Java. Операции (operators) в языке Java. Литералы (константы). Арифметические. Логические литералы. Строковые литералы. Символьные литералы. Операторы (statements). Оператор - выражение. Условный оператор (if). Оператор return (уже рассматривали). Оператор цикла по предусловию (while). Оператор цикла по постусловию (do while). Оператор цикла "со счетчиком" (for). Операторы break и continue ²⁴ . Оператор выбора (switch).	3
4	Массивы в Java	Массивы в Java. Массивы объектов. Многомерные массивы. Присваивание и копирование. Резюмируем основные правила. Конструкторы классов. Вызов одного конструктора из другого. Работа со строками (класс String).	3
5	Структура Java программ	Структура Java программ. Знакомство с библиотеками и пакетами. Использование пакетов в программах. Создание своих собственных пакетов	3
6	Коллекции элементов	Коллекции элементов. Статические вложенные классы. Коллекции объектов Java. Коллекции - это наборы произвольных объектов. Итераторы. Классы реализации коллекций. Коллекции- списки (List).	3
7	Исключительные ситуации	Исключительные ситуации. Обработка исключительных ситуаций (Exceptions). Поведение программы при возникновении исключения. Структура и использование блока перехвата исключений. Классы исключительных ситуаций. Механизм контроля перехвата исключений. Практическая работа. Блок finally. Методы класса Throwable.	3

5.2.2 Практические занятия (семинары)

№ п/п	Наименование раздела дисциплины	Тематика практических занятий (семинаров)	Трудоемкость, в традиционной форме, ак. час
1	Основы Java	Создание собственных классов	2
2	ООП в Java, наследование	Разработка консольных приложений	4
3	Операции в Java	Регулярные выражения	6
4	Массивы в Java	Наследование и коллекции	6
5	Структура Java программ	Работа базой данных	6
6	Коллекции элементов	Графический пользовательский интерфейс	6
7	Исключительные ситуации	Разработка сетевых приложений на Java	6

5.2.3 Лабораторный практикум не предусмотрен.

5.2.4 Самостоятельная работа обучающихся (СРО)

№ п/п	Наименование раздела дисциплины	Вид СРО	Трудоемкость, ак. час
1	2	3	4
1	Основы Java	Проработка материалов по курсу лекций	5
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
2	ООП в Java, наследование	Проработка материалов по курсу лекций	10
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
3	Операции в Java	Проработка материалов по курсу лекций	10
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
4	Массивы в Java	Проработка материалов по курсу лекций	10
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
5	Структура Java программ	Проработка материалов по курсу лекций	8,1
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
6	Коллекции элементов	Проработка материалов по курсу лекций	5
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
7	Исключительные ситуации	Проработка материалов по курсу лекций	5
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	

6 Учебно-методическое и информационное обеспечение дисциплины (модуля)

Для освоения дисциплины обучающийся может использовать:

6.1 Основная литература

1. Гуськова, О. И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. — Москва : МПГУ, 2018. — 240 с. — ISBN 978-5-4263-0648-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/122311> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

2. Березовская, Ю. В. Основы программирования на JAVA: лабораторный практикум : учебно-методическое пособие / Ю. В. Березовская. — Архангельск : САФУ, 2016. — 113 с. — ISBN 978-5-98450-442-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/161794> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

3. Никитенкова, С. П. Многопоточное программирование на языке JAVA : учебно-методическое пособие / С. П. Никитенкова. — Нижний Новгород : ННГУ им. Н. И. Лобачевского, 2015. — 90 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/144990> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

4. Мархакшинов, А. Л. Практикум по программированию на языке Java : учебное пособие / А. Л. Мархакшинов. — Улан-Удэ : БГУ, 2017. — 70 с. — ISBN 978-5-9793-0016-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/154292> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

6.2 Дополнительная литература

1. Кожомбердиева, Г. И. Программирование на языке Java: многопоточные приложения : учебное пособие / Г. И. Кожомбердиева. — Санкт-Петербург : ПГУПС, 2012. — 44 с. — ISBN 978-7641-0401-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/64399> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

2. Макаров, Е. М. Элементы двумерной графики в Java : учебно-методическое пособие / Е. М. Макаров. — Нижний Новгород : ННГУ им. Н. И. Лобачевского, 2015 — Часть 1 — 2015. — 22 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/153092> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

3. Кожомбердиева, Г. И. Криптографическая защита информации и управление доступом на платформе Java : учебно-методическое пособие / Г. И. Кожомбердиева, М. Л. Глухарев. — Санкт-Петербург : ПГУПС, 2016. — 87 с. — ISBN 978-5-7641-0856-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/91082> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

4. Кожомбердиева, Г. И. Использование средств тестирования JUnit при разработке Java-приложений в среде Oracle JDeveloper : учебно-методическое пособие / Г. И. Кожомбердиева, А. М. Сухоногов, Д. А. Протопопов. — Санкт-Петербург : ПГУПС, 2014. — 35 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/49126> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

6.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся

1. Хабитуев, Б. В. Программирование на языке Java: практикум : учебное пособие / Б. В. Хабитуев. — Улан-Удэ : БГУ, 2020. — 94 с. — ISBN 978-5-9793-1548-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/171791> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

2. Строганкова, Н. В. Шаблоны программных платформ языка Java : учебное пособие / Н. В. Строганкова, К. В. Касьяненко, А. В. Хозяинов ; составители практикума входят: использование. — Москва : РТУ МИРЭА, 2021. — 83 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/182466> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

3. Пруцков, А. В. Язык программирования Java. Введение в курс: операторы и типы данных : учебное пособие / А. В. Пруцков. — Рязань : РГРТУ, 2016. — 72 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/168307> (дата обращения: 13.06.2022). — Режим доступа: для авториз. пользователей.

6.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
«Российское образование» - федеральный портал	https://www.edu.ru/
Научная электронная библиотека	https://elibrary.ru/defaultx.asp?
Национальная исследовательская компьютерная сеть России	https://niks.su/
Информационная система «Единое окно доступа к образовательным ресурсам»	http://window.edu.ru/
Электронная библиотека ВГУИТ	http://biblos.vsu.ru/megapro/web
Сайт Министерства науки и высшего образования РФ	https://minobrnauki.gov.ru/
Портал открытого on-line образования	https://npoed.ru/
Электронная информационно-образовательная среда ФГБОУ ВО «ВГУИТ»	https://education.vsu.ru/

6.5 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

При изучении дисциплины используется программное обеспечение и информационные справочные системы: информационная среда для дистанционного обучения «Moodle», автоматизированная информационная база «Интернет-тренажеры», «Интернет-экзамен» и пр.(указать средства, необходимы для реализации дисциплины).

При освоении дисциплины используется лицензионное и открытое программное обеспечение – ОС Windows, MS Office. Локальная сеть университета и глобальная сеть Internet.

7 Материально-техническое обеспечение дисциплины (модуля)

Аудитория для проведения лекционных занятий: ауд.334 - комплект мебели для учебного процесса – 30 шт.; переносной проектор Acer с настольным проекционным экраном, ноутбук Lenovo; наборы демонстрационного оборудования и учебных наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие примерным программам дисциплин рабочим учебным программам.

Аудитории для проведения практических занятий: ауд. 332 – учебная лаборатория для лабораторных и практических работ: количество рабочих станций – 12 (IntelCorei3-540).

Помещения для самостоятельной работы: ауд. 336а - учебная лаборатория для лабораторных, практических работ, курсового и дипломного проектирования: количество рабочих станций – 13 (IntelCorei7- 8700); читальные залы библиотеки: компьютеры со свободным доступом в сеть Интернет и Электронными библиотечными и информационно справочными системами.

8 Оценочные материалы для промежуточной аттестации обучающихся по дисциплине (модулю)

Оценочные материалы (ОМ) для дисциплины (модуля) включают в себя:

- перечень компетенций с указанием индикаторов достижения компетенций, этапов их формирования в процессе освоения образовательной программы;
- описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности.

ОМ представляются в виде отдельного документа и входят в состав рабочей программы дисциплины (модуля) в виде приложения.

Оценочные материалы формируются в соответствии с П ВГУИТ «Положение об оценочных материалах».

ПРИЛОЖЕНИЕ Б
к рабочей программе
Программирование на Java

1. Организационно-методические данные дисциплины для очно-заочной или заочной форм обучения

1.1 Объемы различных форм учебной работы и виды контроля в соответствии с учебным планом

Общая трудоемкость дисциплины (модуля) составляет 4 зачетных единицы

Виды учебной работы	Всего часов	Семестр
		4
	акад.	акад.
Общая трудоемкость дисциплины (модуля)	144	144
Контактная работа, в т.ч. аудиторные занятия:	11,6	11,6
Лекции	4	4
<i>в том числе в форме практической подготовки</i>	-	-
Практические занятия (ПЗ)	4	4
<i>в том числе в форме практической подготовки</i>	4	4
Семинары (С)		
Лабораторные работы (ЛБ)		
Консультации текущие	3,1	3,1
Проведение консультаций перед экзаменом		
Виды аттестации (зачет, экзамен)		экзамен
Самостоятельная работа:	125,6	125,6
Проработка материалов по конспекту лекций	65,6	65,6
Проработка материалов по учебникам, учебным пособиям	60	60
Курсовой проект (работа)		
Реферат		
и (или) другие виды самостоятельной работы		
Подготовка к экзамену (контроль)	6,8	6,8

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

по дисциплине

Программирование на Java

Воронеж

1. Требования к результатам освоения дисциплины

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИД1 _{УК-1} - Анализирует поставленную задачу и осуществляет поиск необходимой информации для ее решения
		ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений
ПКв-8	Способность проводить тестирование компонентов программного обеспечения ИС.	ИД1 _{ПКв-8} - выявление и исправление дефектов и несоответствий в архитектуре и дизайне ИС
		ИД2 _{ПКв-8} - осуществление исправлений дефектов и несоответствий в коде ИС и документации к ИС с последующим подтверждением
		ИД3 _{ПКв-8} - настройка параметров ИС и тестирование результатов настройки
ПКв-9	Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.	ИД1 _{ПКв-9} - использование прикладного инструментария для разработки программного обеспечения в соответствии с назначением объекта проектирования
		ИД2 _{ПКв-9} – демонстрация знаний основ информационного обеспечения прикладных процессов
		ИД3 _{ПКв-9} - разработка программного и информационного обеспечения ИС

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 _{УК-1} - Анализирует поставленную задачу и осуществляет поиск необходимой информации для ее решения	Знает: синтаксис языка Java;
	Умеет: анализировать код программы, написанный на языке Java;
	Владеет: средствами программирования на Java
ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений	Знает: принцип работы с каждым типом данных;
	Умеет: применять систему ввода/вывода на Java
	Владеет: приемами документирования кода программы;
ИД1 _{ПКв-8} - выявление и исправление дефектов и несоответствий в архитектуре и дизайне ИС	Знает: возможности языка и области применения Java – приложений;
	Умеет: создавать консольные приложения на языке Java
	Владеет: приемами работы со встроенными типами данных;
ИД2 _{ПКв-8} - осуществление исправлений дефектов и несоответствий в коде ИС и документации к ИС с последующим подтверждением	Знает: основные типы данных языка Java;
	Умеет: сопоставлять компоненты языка Java
	Владеет: средствами программирования на Java
ИД3 _{ПКв-8} - настройка параметров ИС и тестирование результатов настройки	Знает: основные пакеты, классы, методы языка Java
	Умеет: создавать консольные приложения на языке Java
	Владеет: стандартными методами работы с файлами и каталогами
ИД1 _{ПКв-9} - использование прикладного инструментария для разработки программного обеспечения в соответствии с назначением объекта проектирования	Знает: основной синтаксис языка Java;
	Умеет: анализировать код программы, создавать компоненты языка Java
	Владеет: средствами программирования на Java
ИД2 _{ПКв-9} – демонстрация знаний основ информационного обеспечения прикладных процессов	Знает: принципы создания визуальных форм
	Умеет: работать с базами данных на языке Java
	Владеет: приемами создания визуальных приложений

ИДЗ _{ПКв-9} - разработка программного и информационного обеспечения ИС	Знает: основные визуальные компоненты
	Умеет: создавать консольные приложения на языке Java
	Владеет: технологией работы с базами данных

2. Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые модули/разделы/темы дисциплины	Индекс контролируемой компетенции (или ее части)	Оценочные средства		Технология оценки (способ контроля)
			наименование	№№ заданий	
1	Основы Java ООП в Java, наследование Операции в Java Массивы в Java	УК-1	Тестовые задания	1-168	Рубежный контроль
			Кейс-задания	186-190	Рубежный контроль
			Вопросы к зачету	212-236	Итоговый контроль
			Вопросы к экзамену	237-259	Итоговый контроль
2	Структура Java программ Коллекции элементов Основы Java	ПКв-8	Тестовые задания	1-168	Рубежный контроль
			Кейс-задания	186-190	Рубежный контроль
			Вопросы к зачету	212-236	Итоговый контроль
			Вопросы к экзамену	237-259	Итоговый контроль
3	ООП в Java, наследование Операции в Java Массивы в Java Структура Java программ	ПКв-8	Тестовые задания	1-168	Рубежный контроль
			Кейс-задания	186-190	Рубежный контроль
			Вопросы к зачету	212-236	Итоговый контроль
			Вопросы к экзамену	237-259	Итоговый контроль
4	Коллекции элементов Основы Java ООП в Java, наследование	ПКв-8	Тестовое задание	1-168	Рубежный контроль
			Кейс-задания	186-190	Рубежный контроль
			Вопросы к зачету	212-236	Итоговый контроль
			Вопросы к экзамену	237-259	Итоговый контроль
5	Операции в Java Массивы в Java Структура Java программ Коллекции элементов	ПКв-9	Тестовое задание	1-168	Рубежный контроль
			Кейс-задания	186-190	Рубежный контроль
			Вопросы к зачету	212-236	Итоговый контроль
			Вопросы к экзамену	237-259	Итоговый контроль
6	Исключительные ситуации Основы Java ООП в Java, наследование Операции в Java	ПКв-9	Тестовое задание	1-168	Рубежный контроль
			Кейс-задания	186-190	Рубежный контроль
			Вопросы к зачету	212-236	Итоговый контроль
			Вопросы к экзамену	237-259	Итоговый контроль
7	Массивы в Java, Создание клиентского приложения к СУБД	ПКв-9	Тестовое задание	1-185	Рубежный контроль
			Кейс-задания	186-211	Рубежный контроль
			Вопросы к зачету	212-236	Итоговый контроль
			Вопросы к экзамену	237-259	Итоговый контроль

3. Оценочные средства для промежуточной аттестации

3.1 Тесты (тестовые задания)

3.1.1 УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

№ задания	Тестовое задание
1.	Дан код: <code>class Quest3 {public static void main(String s[]) { String args; System.out.print(args + s);}}</code> Результатом компиляции кода будет? >> ошибка компиляции: переменная <i>args</i> используется до инициализации.
2.	Как получить объект класса Color из AWT, описывающий чистый синий цвет? 1) new Color("blue") 2) new Color(0, 0, 255) 3) Color.getBlue() 4) Color.blue
3.	Какие методы есть в классе Object? 1) public boolean equals(Object obj) 2) public int notify() 3) public int hashCode() 4) protected Object clone() 5) public boolean wait()
4.	Дан код: <pre>public class Quest4 { public static void main(String[] args) { byte b[] = new byte[80]; for (int i=0; i<b.length; i++) b[i] = (byte) System.in.read(); System.out.print("Ok"); } }</pre> Результатом компиляции запуска будет? >> ошибка компиляции, так как метод read() может породить исключительную ситуацию типа IOException.
5.	Программы Java категория Приложение (application)? >> аналог "обычной" прикладной программы.
6.	Программы Java категория Апплет (applet)? >> специализированная программа с ограниченными возможностями, работающая в окне WWW-документа под управлением браузера.
7.	Программы Java категория Сервлет (servlet)? >> специализированная программа с ограниченными возможностями, работающая в WWW на стороне сервера. Используется преимущественно в рамках технологии JSP для программирования WWW-документов со стороны сервера.
8.	Программы Java категория Серверное приложение (Enterprise application)? >> предназначено для многократного использования на стороне сервера.
9.	Программы Java категория Библиотека (Java Class Library)? >> библиотека классов, либо NetBeansModule – модуль платформы NetBeans) – предназначена для многократного использования программами Java.
10.	Какие из следующих строк скомпилируются без ошибки? >> int i = 32565; - float f = 7.0; - char c = "z"; - byte b = 255; - boolean n = null;
11.	Какие из следующих строк скомпилируются без ошибки? >> int j = 'ъ'; - float f = 7.0; - char c = "z"; - byte b = 255; - boolean n = null;

12.	Какие варианты записи оператора условного перехода корректны? >> if (i<j) { System.out.print("-1-"); } - if (i<j) then System.out.print("-2-"); - if i<j { System.out.print("-3-"); } - if [i<j] System.out.print("-4-"); - if {i<j} then System.out.print("-6-");
13.	Какие варианты записи оператора условного перехода корректны? >> if (i<j) System.out.print("-5-"); - if (i<j) then System.out.print("-2-"); - if i<j { System.out.print("-3-"); } - if [i<j] System.out.print("-4-"); - if {i<j} then System.out.print("-6-");
14.	Какие из следующих идентификаторов являются корректными? >> _int; - 2int; - int #; - #int - _#int
15.	Назначение утилиты javac? >> Компилятор в режиме командной строки для программ, написанных на языке Java.
16.	Назначение утилиты java? >> Утилита для запуска в режиме командной строки откомпилированных программ-приложений.
17.	Назначение утилиты appletviewer? >> Утилита для запуска на исполнение и отладку апплетов без браузера. При этом не гарантируется работоспособность отлаженного апплета в браузере.
18.	Назначение утилиты jdb? >> Отладчик программ, написанных на языке Java.
19.	Назначение утилиты javadoc? >> Генератор документации по классам на основе комментариев, начинающихся с /**.
20.	Назначение утилиты jar? >> Создание и управление Java-архивами jar.
21.	Назначение утилиты javah? >> Генератор заголовочных файлов C/C++ для подключения к программам Java внешних библиотек C/C++ на основе интерфейса JNI.
22.	Назначение утилиты javap? >> Дизассемблер классов.
23.	Назначение утилиты extcheck? >> Утилита для обнаружения конфликтов между файлами архивов jar.
24.	Назначение утилиты native2ascii? >> Утилита для конвертации в режиме командной строки параметра, передаваемого в виде текста на национальном алфавите, в последовательность символов UNICODE.
25.	Управляющая последовательность: \a ? >> звонок.
26.	Управляющая последовательность: \b ? >> возврат на один символ назад.
27.	Управляющая последовательность: \f ? >> перевод на новую страницу.
28.	Управляющая последовательность: \n ? >> перевод на новую строку.
29.	Управляющая последовательность: \r ? >> возврат к началу строки.
30.	Управляющая последовательность: \t ? >> горизонтальная табуляция.
31.	Управляющая последовательность: \v ? >> вертикальная табуляция.
32.	Управляющая последовательность: \' ? >> кавычка.

33.	Управляющая последовательность: >> двойные кавычки.	"	?
34.	Управляющая последовательность: >> обратная косая черта.	\"	?
35.	Управляющая последовательность: >> начало кодировки символа Unicode.	\u	?
36.	Оператор >> Оператор побитового дополнения (побитовое "не").	~	?
37.	Оператор >> Оператор "исключающее или" (XOR).	^	?
38.	Оператор >> Оператор "побитовое и" (AND).	&	?
39.	Оператор >> Оператор "побитовое или" (OR).		?
40.	Оператор >> Оператор "НЕ".	!	?

3.1.2 ПКв-8 Способность проводить тестирование компонентов программного обеспечения ИС.

№ задания	Тестовое задание
41.	Оператор && >> Оператор "логическое И" (AND).
42.	Оператор >> Оператор "логическое ИЛИ" (OR).
43.	Оператор << >> Оператор левого побитового сдвига.
44.	Оператор >>> >> Оператор беззнакового правого побитового сдвига.
45.	Оператор >> >> Оператор правого побитового сдвига с сохранением знака отрицательного числа.
46.	Для чего предназначен пакет java.applet? >> Классы, необходимые для создания апплетов.
47.	Для чего предназначен пакет java.awt? >> Графический пользовательский интерфейс на базе AbstractWindowsToolkit.
48.	Для чего предназначен пакет java.awt.color? >> Пространства цветов для AWT.
49.	Для чего предназначен пакет java.awt.datatransfer? >> Передача данных внутри программы и между программами.
50.	Для чего предназначен пакет java.awt.dnd? >> Поддержка технологии drag-and-drop.
51.	Для чего предназначен пакет java.awt.event? >> Обработка событий AWT.
52.	Для чего предназначен пакет java.awt.im? >> Поддержка нестандартных методов ввода текста.
53.	Для чего предназначен пакет java.awt.im.spi? >> Интерфейсы для создания нестандартных методов ввода текста.
54.	Для чего предназначен пакет java.lang? >> Базовые классы языка Java.
55.	Для чего предназначен пакет java.lang.ref? >> Управление ссылками на объекты.
56.	В папке build проекта NetBeans находятся? >> хранятся скомпилированные файлы классов, имеющие расширение .class.
57.	В папке dist проекта NetBeans находятся? >> файлы, предназначенные для распространения как результат компиляции (модуль JAR приложения или библиотеки, а также документация к нему).

58.	В папке nbproject проекта NetBeans находятся? >> находится служебная информация по проекту.
59.	В папке src проекта NetBeans находятся? >> исходные коды классов. Кроме того, там же хранится информация об экранных формах (которые будут видны на экране в виде окон с кнопками, текстом и т.п.). Она содержится в XML-файлах, имеющих расширение .form.
60.	В папке test проекта NetBeans находятся? >> сопроводительные тесты, предназначенные для проверки правильности работы классов проекта.
61.	Команды документации (символы метаданных) @see? >> применяется для создания в документе гиперссылок на другие комментарии. Можно использовать для любых конструкций (классов, методов и т.д.).
62.	Команды документации (символы метаданных) @since? >> Информация о версии JDK, начиная с которой введён или работоспособен класс или интерфейс.
63.	Команды документации (символы метаданных) @throws? >> Информация об исключительных ситуациях, которые могут возбуждаться методом.
64.	Команды документации (символы метаданных) @param? >> информация о параметре метода.
65.	Команды документации (символы метаданных) @deprecated? >> информация о том, что данный метод устарел и в последующих версиях будет ликвидирован. При попытке использования таких методов компилятор выдаёт программисту предупреждение (warning) о том, что метод устарел, хотя и компилирует проект.
66.	Целые типы, тип byte? >> Однобайтовое целое число (8-битное целое со знаком).
67.	Целые типы, тип short? >> Короткое целое число (16- битное целое со знаком).
68.	Целые типы, тип char? >> Символьный тип (беззнаковое 16- битное целое).
69.	Целые типы, тип int? >> Целое число (32- битное целое со знаком).
70.	Целые типы, тип long? >> Длинное целое число (64- битное целое со знаком).
71.	Выберите правильный вариант a,b,c,d чтобы результат выражения a&&b c^ d являлся истиной? >> все ответы верны - a=true, b=true, c=false, d=true - a=false, b=false, c=false, d=true - a=false, b=false, c=true, d=false - a=true, b=true, c=true, d=true
72.	Выберите правильный вариант a,b,c,d чтобы результат выражения a&&(b c)^ d являлся ложью? >> все ответы верны - a=false, b=true, c=false, d=true - a=true, b=true, c=false, d=true - a=false, b=false, c=false, d=true - a=false, b=false, c=false, d=false 71. Вещественные типы, размер типа float? >> 4 байта
73.	Вещественные типы, размер типа double? >> 8 байт

3.1.3 ПКв-9 Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

№ задания	Тестовое задание
74.	<pre>int i = 0; i++; System.out.print(i); i = i++; System.out.println(i);</pre> <p>Что выведет данный код? >> 11</p>
75.	<pre>int i = 0, j = 5; tp: for (;;) { i++; for (;;) { if (i > --j) { breaktp; } } System.out.println("i = " + i + ", j = " + j);}</pre> <p>Что произойдет во время компиляции и выполнения данного фрагмента кода? >> Ошибка компиляции.</p>
76.	<pre>int i = 1; //1 i = -(10 + 2 + i); //2 ++i--; //3 System.out.println(i);</pre> <p>В какой строке(-ах) кода содержится ошибка? >> 3.</p>
77.	<pre>Integer i = new Integer("10"); if (i.toString().intern() == i.toString().intern()){ System.out.println("Равный"); } else { System.out.println("Не равный"); }</pre> <p>Результат выполнения программы? >> Выведется "Равный".</p>
78.	<pre>public class Main { public static void main(String[] args) { Boolean b1 = new Boolean("true"); System.out.print(b1 + " "); b1 =new Boolean("tRuE"); System.out.print(b1 + " "); b1 =new Boolean("test"); System.out.print(b1 + " "); b1 =new Boolean(true); System.out.print(b1 + " "); b1 = true; System.out.println(b1); }}</pre> <p>каким будет результат компиляции и выполнения кода? >> true true false true true</p>
79.	<pre>Integer i = new Integer("10"); if (i.toString().intern() == i.toString().intern()){ System.out.println("Равный"); } else { System.out.println("Не равный"); }</pre> <p>Результат выполнения программы? >> Выведется "Равный".</p>
80.	<pre>public class Main { public static void main(String[] args) { Boolean b1 = new Boolean("true"); System.out.print(b1 + " "); b1 =new Boolean("tRuE"); System.out.print(b1 + " "); b1 =new Boolean("test"); System.out.print(b1 + " "); b1 =new Boolean(true); System.out.print(b1 + " "); b1 = true; System.out.println(b1); }}</pre> <p>каким будет результат компиляции и выполнения кода? >> true true false true true</p>
81.	<pre>public class Test { static { System.out.print("Bo-Bo ");} public static void main(String[] args) { System.out.print("Hello, World! "); } static { System.out.println("Go-Go "); System.exit(0); }}</pre> <p>что выведет код? >> Bo-Bo Go-Go</p>

82.	<pre>int i = 1; //1 i = -(10 + 2 + 1); //2 ++i--; //3 System.out.println(i);</pre> <p>В какой строке(-ах) кода содержится ошибка? >> 3</p>
83.	<pre>Integer a = 128; Integer b = 128; Integer c = -128; Integer d = -128; System.out.print((a == b)+" "); System.out.println(c == d);</pre> <p>Что будет выведено на экран? >> false true</p>
84.	<pre>Boolean b = new Boolean("Hello"); System.out.println(b);</pre> <p>Какой результат выполнения данного кода? >> false</p>
85.	<pre>public class Test { static { i =5; } static int i = 6; public static void main(String[] args) { System.out.println(i); }</pre> <p>Выберите правильный ответ? >> 6</p>
86.	<pre>public class MathTest { public static void main(String[] args) { System.out.print(((-7 % 5) > (7 % -5)) + " "); System.out.println(Math.abs(Math.ceil(-2.5)) < Math.ceil(Math.abs(-2.5))); }</pre> <p>Что будет напечатано в результате выполнения данного кода? >> false true</p>
87.	<pre>int i = 0; System.out.print(i++); System.out.println(i);</pre> <p>Результат кода? >> 01</p>
88.	<pre>int j = 0; System.out.print(++j); System.out.println(j);</pre> <p>Результат кода? >> 11</p>
89.	<pre>public class TestClass { int i = getInt(); int k = 20; public int getInt(){ return k+1;} public static void main(String[] args) { TestClass t = new TestClass (); System.out.println(t.i+" "+t.k); }</pre> <p>Что выведет этот код? >> Будет выведено на печать 1 20</p>
90.	<pre>. int x = 0; int y = 10; do{ y--; ++x;} while(x<5); System.out.print(x+" "+y); }</pre> <p>Что напечатает этот код? >> напечатает 5 5</p>
91.	<p>Дана строка: String s="true,false"; Каким способом можно сформировать новую строку, равную "test"? >> String n = s.substring(0,1) + s.substring(9,10) + s.substring(8,9) + s.substring(0,1)</p>
92.	

93.	<pre>String s1="Строка типа String"; String s2="Строка"; s2+=" типа String"; if(s1==s2) System.out.println("s1 равно s2"); else System.out.println("s1 не равно s2"); Что выведет этот код? >> s1 не равно s2</pre>
94.	<pre>String s1="Строка типа String"; String s2="Строка"; s2+=" типа String"; if(s1.equals(s2)) System.out.println("s1 равно s2"); else System.out.println("s1 не равно s2"); Что выведет этот код? >> s1 равно s2</pre>
95.	<pre>String s=""; s+= 1+2+3; System.out.println(s); Что выведет этот код? >> 6</pre>
96.	<pre>String s="Сумма =" +1+2+3; System.out.println(s); Что выведет этот код? >> Сумма=123</pre>
97.	<pre>String s=1+2+" не равно "+1+2; System.out.println(s); Что выведет этот код? >> 3 не равно 12</pre>
98.	<p>Работа с графикой. Функция <i>clearRect(x,y,w,h)?</i> >> Очистка прямоугольника (заполнение текущим цветом).</p>
99.	<p>Работа с графикой. Функция <i>fill3DRect(x,y,w,h, isRaised)?</i> >> Вывод «объемного» прямоугольника, заполненного текущим цветом.</p>
100.	<p>Работа с графикой. Функция <i>fillPolygon(Polygon p)?</i> >> Вывод многоугольника, заполненного текущим цветом.</p>
101.	<p>Работа с графикой. Функция <i>copyArea(x,y,w,h,dx,dy)?</i> >> Копирование области на новое место.</p>
102.	<p>Переменные класса <i>File.String pathSeparator?</i> >> Содержит строку с символом разделителя пути в операционной системе. Это «/» в Unix-подобных системах и «\» в Windows.</p>
103.	<p>Переменные класса <i>File.char pathSeparatorChar?</i> >> Содержит символ разделителя пути в операционной системе. Это «/» в Unix-подобных системах и «\» в Windows.</p>
104.	<p>Переменные класса <i>File.String separator?</i> >> Содержит строку с символом разделителя между именами файлов и файловых масок в операционной системе.</p>
105.	<p>Методы класса <i>File.canRead() ?</i> >> Возвращает true в случае, когда файл (или папка) с заданным в конструкторе именем существует и доступен по чтению. Иначе false.</p>

106.	<p>Методы класса <i>File.setReadOnly()</i> ?</p> <p>>> Возвращает true в случае, когда файл (или папка) с заданным в конструкторе именем существует, и ему удалось установить статус «доступен только по чтению». Иначе false.</p>
107.	<p>Методы класса <i>File.isDirectory()</i> ?</p> <p>>> Возвращает true в случае, когда файл или папка с заданным в конструкторе именем существует и является папкой. Иначе false.</p>
108.	<p>Методы класса <i>File.isFile()</i> ?</p> <p>>> Возвращает true в случае, когда файл или папка с заданным в конструкторе именем существует и является файлом. Иначе false.</p>
109.	<p>Методы класса <i>File.isHidden()</i> ?</p> <p>>> Возвращает true в случае, когда файл или папка с заданным в конструкторе именем существует и является скрытым. Иначе false.</p>
110.	<p>Методы класса <i>File.lastModified()</i> ?</p> <p>>> Возвращает время последней модификации файла, если он существует и доступен по чтению. Иначе 0L.</p>
111.	<p>Методы класса <i>File.setLastModified(long time)</i> ?</p> <p>>> Устанавливает время последней модификации файла. Возвращает true, если он существует и доступен по записи. Иначе false.</p>
112.	<p>Методы класса <i>File.getName()</i> ?</p> <p>>> Возвращает короткое имя файла или папки</p>
113.	<p>Методы класса <i>File.getAbsolutePath()</i> ?</p> <p>>> Возвращает абсолютный путь к файлу или папке, включая имя файла. При этом, если в имени файла в конструкторе была задана относительная адресация, соответствующая часть пути сохраняется в возвращаемой строке.</p>
114.	<p>Методы класса <i>File.getCanonicalPath()</i> ?</p> <p>>> Возвращает абсолютный путь к файлу или папке, включая имя файла. При этом, если в имени файла в конструкторе была задана относительная адресация, соответствующая часть пути заменяется в возвращаемой строке на канонический вариант адресации – без элементов относительной адресации.</p>
115.	<p>Методы класса <i>File.compareTo(File f)</i> ?</p> <p>>> Сравнение имён файлов (папок), сопоставляемых текущему файловому объекту и объекту f. Возвращает 0 в случае, когда абсолютные имена файлов (папок) совпадают. Иначе возвращает число, зависящее от разницы в длинах имён и кодов, составляющих их символов.</p>
116.	<p>Методы класса <i>File.isAbsolute()</i> ?</p> <p>>> Возвращает true в случае, когда адресация к имени файла (папки) текущего файлового объекта является абсолютной. Хотя может содержать элементы относительной адресации, то есть не быть канонической.</p>
117.	<p>Методы класса <i>File.equals(Object obj)</i> ?</p> <p>>> Возвращает true тогда и только тогда, когда текущий объект и параметр obj соответствует одному и тому же файлу (папке).</p>
118.	<p>Методы класса <i>File.createTempFile(String prefix, String suffix)</i> ?</p> <p>>> Обеспечивает создание пустого файла (или папки), задаваемого коротким именем prefix+suffix в папке операционной системы, предназначенной для временных файлов. Возвращает ссылку на объект. Префикс должен быть не менее 3 символов.</p>
119.	<p>Методы класса <i>File.mkdir()</i> ?</p> <p>>> Попытка создания папки по имени, которое было задано в конструкторе объекта. Возвращает true в случае успешного создания и false в других случаях.</p>
120.	<p>Методы класса <i>File.renameTo(File dest)</i> ?</p> <p>>> Попытка переименования файла или папки с имени, которое было задано в конструкторе объекта, на новое, задаваемое параметром dest. Возвращает true в случае успешного переименования и false в других случаях.</p>
121.	<p>Методы класса <i>File.getAbsolutePathFile()</i> ?</p> <p>>> Создание нового файлового объекта по абсолютному пути, соответствующему текущему файловому объекту.</p>

122.	Методы класса <i>File.list()</i> ? >> Возвращает массив строк (список) коротких имён, находящихся в папке файлов и папок. Имена элементов, находящихся во вложенных папках, не показываются. Если файловый объект не соответствует существующей папке, возвращает null.
123.	Конструкторы? >> это методы, используемые для инициализации объектов в процессе их реализации.
124.	Как объекты обмениваются информацией? >> Объекты общаются друг с другом посредством посылки сообщений.
125.	Что такое <i>this</i> ? >> ссылка, с помощью которой объект обращается к самому себе.
126.	Класс ? >> Класс – это структурный тип данных, который включает описание полей данных, а также процедур и функций, работающих с этими полями данных.
127.	Процедуры и функции, работающие с полями данных класса? >> Называются методами.
128.	Объединение полей данных и процедур и функций, работающих с этими данными, в единый пакет, при наличии специальных правил доступа к элементам пакета? >> Инкапсуляция.
129.	Основные средства разработки классов. Наследование? >> Конструирование новых более сложных классов из уже имеющихся посредством добавления полей и определения новых методов (принцип иерархичности).
130.	Основные средства разработки классов. Полиморфизм? >> обеспечивает возможность задания различных реализаций некоторого единого по названию метода для классов различных уровней иерархии.
131.	Полиморфными объектами или полиморфными переменными называются? >> переменные, которым в процессе выполнения программы может быть присвоено значение, тип которого отличается от типа переменной.
132.	К изменению свойств объекта возможен только через принадлежащие ему? >> методы.
133.	В момент времени объект характеризуется?: >> Свойствами
134.	Программа в объектно-ориентированном программировании состоит из?: >> событийных процедур.
135.	Инкапсулированные в классе процедуры и функции называются? >> Методы
136.	Процесс, при котором различный код и данные различных модулей соединяются друг с другом в процессе исполнения, называют компоновкой? >> Динамической
137.	Область исходного кода программы, из которой допустим нормальный доступ к связанному с идентификатором объекту, называется? >> Видимостью
138.	Конструктором по умолчанию называется такой конструктор, который? >> не принимает никаких аргументов.
139.	Класс-наследник называется ... от базового класса? >> Производным.
140.	Что следует ожидать на экране? System.out.print("A"+"B"+"C"); System.out.println('A'+ 'B'+ 'C'); >> ABC198

141.	<pre>. int x=0; for(int count=10; count < 5; count++){ x += count ;} Чему равна x? >> 0</pre>
142.	<p>Каков будет результат следующего примера?</p> <pre>Point p = new Point(1,2); int a =p.x; p = new Point(3,4); System.out.println(a); >> 1</pre>
143.	<p>Какие из следующих утверждений истинные?</p> <p>a-nonstatic-метод не может быть вызван из статического метода;</p> <p>b-static-метод не может быть вызван из нестатического метода;</p> <p>c-private-метод не может быть вызван из другого метода этого класса;</p> <p>d-final-метод не может быть статическим.</p> <p>>> a</p>
144.	<p>Какие из объявлений корректны, если</p> <pre>class Owner{ class Inner{ }}? >> new Owner().new Inner();</pre>
145.	<p>Экземпляром какого класса является поле <i>System.in</i>?</p> <p>>> java.io.InputStream;</p>
146.	<p>Какой абстрактный класс является суперклассом для всех классов, используемых для чтения байтов?</p> <p>>> InputStream</p>
147.	<p>Какой интерфейс наиболее пригоден для создания класса, содержащего несORTиРОВанные уникальные объекты?</p> <p>>> Set</p>

3.2 Кейс-задания.

3.2.1 УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

№ задания	Формулировка задачи
148.	<pre>Float f1 = new Float(Float.NaN); Float f2 = new Float(Float.NaN); System.out.println(""+ (f1 == f2)+ " "+f1.equals(f2)+ " "+(Float.NaN == Float.NaN));</pre> <p>Что будет выведено в результате выполнения данного куска кода? >> false true false</p>
149.	<p>Дан код: <applet code=MyApplet.class width=200 height=200> <param name=count value=5> </applet></p> <p>Какой код читает параметр count в переменную i? >> int i = new Integer(getParameter("count")).intValue() Источник: http://yznaika.com/notes/629-java-answers</p>
150.	<p>Дан код: <pre>public class Quest1 { public static void main(String[] args) { String str = new String("java"); int i=1; char j=3; System.out.println(str.substring(i,j));}</pre></p> <p>В результате при компиляции запуске будет выведено? >></p>
151.	<p>Дан код: <pre>public class Quest5 { {System.out.print("1");} static{System.out.print("2");} Quest5(){System.out.print("3");} public static void main(String[] args) { System.out.print("4"); }}</pre></p> <p>В результате при компиляции и запуске будет выведено? >> 24</p>
152.	<p>Дан код: <pre>public class Quest5{ public static void main(){ System.out.print("A"); } public static void main(String args){ System.out.print("Б"); } public static void main(String[] args){ System.out.print("B");}}</pre></p> <p>Что будет выведено в результате компиляции и запуска? >> B</p>
153.	<p>Дан код: <pre>class Quest3 { public static void main(String s[]) { String args; System.out.print(args + s); }}</pre></p> <p>Результатом компиляции кода будет? >> ошибка компиляции: переменная args используется до инициализации.</p>

154.	<p>Каков будет результат следующего примера?</p> <pre>int b=5; System.out.print(b+(b=3)); System.out.print((b=3)+3);b=5; System.out.print(b+=(b=3)); >> 868</pre>
155.	<p>Каков будет результат следующей программы?</p> <pre>public class testClass { public static void main(String[] args) { inti =5; while(i=5){ System.out.println(i++); }}</pre> <p>>> ОШИБКА КОМПИЛЯЦИИ.</p>
156.	<p>Каков будет результат следующего примера?</p> <pre>class Parent{static int a;} class Child extends Parent{ } public class testClass { public static void main(String[] args) { Child.a=10; Parent.a=5; System.out.println(Child.a); }}</pre> <p>>> 5</p>
157.	<p>Каков будет результат следующего примера?</p> <pre>int i,j; lab:for(i=0;i<6;i++){ for(j=3;j>1;j--){ if(i==j){System.out.print(j);break lab;} } }</pre> <p>>> 2</p>
158.	<p>Каков будет результат следующего примера?</p> <pre>class Parent{int x =0; public void printX(){System.out.print(x);}} class Child extends Parent{ int x = -1;} public class testClass { public static void main(String[] args) { new Child().printX(); }}</pre> <p>>> 0</p>
159.	<p>Каков будет результат следующего примера?</p> <pre>class Parent{int a =2;} class Child extends Parent{ int a = 3;} public class testClass { public static void main(String[] args) { Child c = new Child(); System.out.print(c.a); Parent p=c; System.out.print(p.a); }}</pre> <p>>> 32</p>
160.	<p>Каков будет результат следующих действий?</p> <pre>Point p1 = new Point(2,3); Point p2 = new Point(2,3); System.out.print(p1==p2); p2=p1; p1.x = 3; System.out.print(" "); System.out.println(p1==p2); >> false true</pre>
161.	<p>Каков будет результат следующего примера?</p> <pre>public class testClass { public static void process (int x) {x = 5;} public static void main(String[] args) {</pre>

	<pre>int x=3; process(x); System.out.println(x); }} >> 3</pre>
162.	<pre>class A{public A() {System.out.print("Создание класса А. ");}} class B extends A{ public B(){System.out.print("Создание класса В. ");}} class C extends B{public C(){System.out.print("Создание класса С. ");}} public class testClass { public static void main(String[] args) { C c = new C(); }} Что выведется в результате? >> Создание класса А. Создание класса В. Создание класса С.</pre>
163.	<p>Что будет напечатано в консоли в результате выполнения следующего кода:</p> <pre>System.out.println((byte)383+" "+(byte)384+" "+(byte)-384); ? >> 127 -128 -128</pre>
164.	<p>Каков будет результат следующего примера?</p> <pre>Point p1 = new Point(3,4); Point p2 = p1; p1.x = 5; p1 = new Point(4,4); System.out.println(p2.x-p1.x); >> 1</pre>
165.	<pre>int result = 12+2*5%3-15/4; String x=12-6+"Hello"+7+5; System.out.println(result+"-"+x); Что будет выведено на консоль? >> 10-6Hello75</pre>
166.	<pre>int x, y; int i; x = 1; for (i = 0; i < 10; i++) { y = x + ++x; System.out.print(y+" "); } Что выведется в результате? >> 3 5 7 9 11 13 15 17 19 21</pre>
167.	<pre>int x, y; int i; x = 1; for (i = 0; i < 10; i++) { y = x + x++; System.out.print(y+" "); } Что выведется в результате? >> 2 4 6 8 10 12 14 16 18 20</pre>
168.	<pre>for(int i=-10; i<= 10; i++) {if(i > 0) break; System.out.print(i + " ");}, Что выведется в результате? >> -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0</pre>
169.	<pre>int j = 0; for (int i = 0; i < 10; i++) j = j++; System.out.println(j); Что выведется в результате? >> 0</pre>
170.	<p>Каков будет результат следующего примера?</p> <pre>char c = 65; System.out.print(c); System.out.print(+c); System.out.print("="+c); >> A65=A</pre>
171.	<pre>String s1="Строка"; String s2="Строка"; if(s1==s2) System.out.println("s1 равно s2");</pre>

	<pre>else System.out.println("s1 не равно s2");</pre> <p>Что выведет этот код? >> s1 равно s2</p>
172.	<p>Есть следующий код: <code>public class SomeClass { public static void main(String[] args) { SomeClass t = new SomeClass(); ____ } private int check(String x, int n) { if (n == 0) return n; else if (n == 1) { if (x != null) return 5; } else if (n == 2 && x != null) { if (x.equals("YES")) return 3; else if (x.equals("NO")) return 4; } return -1; } }</code> Если вместо знаков подчеркивания поместить код вызова метода <code>check</code>, то какое из предложений будет верным?</p> <ol style="list-style-type: none"> 1) <code>t.check("ANY",1)</code>, в этом случае функция вернет значение 14 2) <code>t.check("NO",2)</code>, в этом случае функция вернет значение 4 3) <code>t.check("YES",1)</code>, в этом случае функция вернет значение 3 4) <code>t.check("YES",2)</code>, в этом случае функция вернет значение 4 5) <code>t.check("ANY",6)</code>, в этом случае функция вернет значени
173.	<p>Дан следующий класс: <code>class abcPoint { abcPoint(int a, int b) { x = a; y = b; } int x, y; ... }</code> Укажите результат выполнения следующего фрагмента кода: <code>abcPoint p = new abcPoint(5,3), p1 = p; p.x = 4; System.out.println(p1.x);</code></p> <ol style="list-style-type: none"> 1) 3 2) 4 3) 5 4) будет выдано сообщение об ошибке 5) ничего из перечисленного
174.	<p>Будет ли скомпилирован следующий пример: <code>byte b = 100 - 100; b = 100 + 200; b = 100 * 100;</code></p> <ol style="list-style-type: none"> 1) пример не будет скомпилирован, будет выдано сообщение об ошибках в 1-й и 2-й строках 2) пример не будет скомпилирован, будет выдано сообщение об ошибках во 2-й и 3-й строках 3) пример не будет скомпилирован, будет выдано сообщение об ошибках в 1-й и 3-й строках 4) пример не будет скомпилирован, будет выдано сообщение об ошибках в 1-й, 2-й и 3-й строках 5) пример будет скомпилирован

3.2.2 ПКв-8 Ис. Способность проводить тестирование компонентов программного обеспечения

№ задания	Формулировка задачи
175.	<p>Что будет выведено на экран в результате выполнения следующего кода: <code>public class SomeClass { public static void main(String[] args) { int i = 0; while (i < 6) { if (i > 5) i--; if (i < 4) i++; if (i == 5) i++; i++; System.out.print(i); } }</code></p> <ol style="list-style-type: none"> 1) 1234 2) 245 3) 2457 4) 3457 5) ошибка времени выполнения
176.	<p>Укажите результат выполнения следующего кода: <code>public abstract class SomeClass implements Runnable { public final static Object one = new Object(), two = new Object(); public static void main(String s[]) { Thread t1 = new Thread() { public void run() { synchronized (one) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } synchronized (two) { System.out.print("1"); } } } }; Thread t2 = new Thread() { public void run() { synchronized (two) { synchronized (one) { System.out.print("2"); } } } }; t1.start(); t2.start(); }</code></p> <ol style="list-style-type: none"> 1) программа не завершит работу, на консоли появится 1 2) программа не завершит работу, на консоли появится 2 3) программа не завершит работу

	4) программа не завершит работу, на консоли появится 12
177.	Какой тип должна иметь переменная d в следующем фрагменте: int a = 1; short b = 1; long c = 1; ? d = a + b + c; 1) byte 2) short 3) int 4) long 5) char
178.	Пусть классы Wolf и Rabbit являются наследниками класса Animal. Корректен ли следующий пример? Wolf w = new Wolf(); Animal a = (Animal)w; Rabbit r = (Rabbit)a; 1) пример не будет откомпилирован, ошибка в первой строке 2) пример не будет откомпилирован, ошибка во второй строке 3) пример будет откомпилирован, но возникнет ошибка времени исполнения 4) пример полностью корректен
179.	Какой метод сгенерирует ошибку ArrayStoreException при передаче в него массива v? public class Vehicle { public static void main(String[] args) { Vehicle[] v = new Car[5]; ... } public void setVehicles(Vehicle c[]) { c[0] = new Vehicle(); } public void setVehicles2(Vehicle c[]) { if (c[0] instanceof Vehicle) { c[0] = new Vehicle(); } } public void setVehicles3(Car c[]) { if (c[0] instanceof Car) { c[0] = new Car(); } } } class Car extends Vehicle{} 1) setVehicles 2) setVehicles2 3) setVehicles3
180.	Есть следующая иерархия наследования: java.lang.Object +--java.lang.Throwable +-java.lang.Exception +--java.lang.RuntimeException +--java.lang.IndexOutOfBoundsException +--java.lang.StringIndexOutOfBoundsException +--java.lang.ArrayIndexOutOfBoundsException В методе testSomeValue могут быть возбуждены исключения StringIndexOutOfBoundsException и ArrayIndexOutOfBoundsException, при этом они не обрабатываются в блоке try – catch. Какое из перечисленных ниже утверждений будет верным? 1) определение метода testSomeValue должно включать throws StringIndexOutOfBoundsException, ArrayIndexOutOfBoundsException 2) если метод, вызывающий testSomeValue, перехватывает IndexOutOfBoundsException, то исключения StringIndexOutOfBoundsException, ArrayIndexOutOfBoundsException тоже будут перехватываться 3) так как в определении метода указано throws StringIndexOutOfBoundsException, ArrayIndexOutOfBoundsException, любой вызывающий его метод должен перехватывать эти типы исключений, вне зависимости от того, возбуждается во время работы исключение или нет 4) при объявлении метода testSomeValue необходимо указывать возбуждаемые исключения
181.	Даны следующие классы: class abc.SomeClass {} class abc.SomeClass2 extends abc.SomeClass {} Что будет выведено на экран в следующем фрагменте кода: abc.SomeClass x1 = new abc.SomeClass(); abc.SomeClass2 x2 = new abc.SomeClass2(); System.out.println((x1 instanceof abc.SomeClass) + " " + (x2 instanceof abc.SomeClass)); 1) false false 2) true false 3) true true 4) false true
182.	Корректен ли объявленный ниже класс? Если нет, то как его можно исправить? class Box { private int weight=0; public int getWeight() { return weight; } void setWeight(int weight) { weight=weight; } } 1) пример корректен, ничего исправлять не нужно 2) пример некорректен, конфликт имен в методе setWeight требует переименования поля или аргумента 3) пример корректен, однако конфликт имен в методе setWeight требует переименования поля или аргумента 4) пример корректен, однако конфликт имен в методе setWeight требует переименования поля или аргумента, либо использования ключевого слова this 5) пример корректен, однако конфликт имен в методе setWeight требует

	переименования поля или аргумента, либо использования ключевого слова this или super
--	--

3.2.3 ПКв-9 Способность осуществлять ведение базы данных и поддержку информационно-го обеспечения решения прикладных задач.

№ задания	Формулировка задачи
183.	<p>Укажите результат выполнения следующего кода: <code>public class SomeClass { public static void main(String[] args) { SomeClass c = new SomeClass(); c.f1(1, 1); } public void f1(byte b, char c) { System.out.println("1"); } public void f1(byte b, byte c) { System.out.println("2"); } public void f1(char b, char c) { System.out.println("3"); } public void f1(char b, byte c) { System.out.println("4"); } }</code></p> <p>1) 1 2) 2 3) 3 4) 4 5) возникнет ошибка компиляции</p>
184.	<p>Какой вариант кода выведет на экран значение true?</p> <p>1) <code>int x=0; System.out.println(++x==++x);</code> 2) <code>int x=0; System.out.println(++x==x);</code> 3) <code>int x=0; System.out.println(x==x++);</code> 4) <code>int x=0; System.out.println(x++==x++);</code> 5) <code>int x=0; System.out.println(x++==++x);</code></p>
185.	<p>Какое значение появится на консоли после выполнения следующей программы? <code>class Parent { int x = 2; public void print() { System.out.println(x); } } class Child extends Parent { int x = 3; public static void main(String[] args) { new Child().print(); } public void print(String val){ System.out.println(val); } }</code></p> <p>1) 0 2) 1 3) 2 4) 3 5) будет выведено сообщение об ошибке</p>
186.	<p>Что будет выведено на экран в результате выполнения следующего кода: <code>public class Test { public static void main(String[] args) { Test t = new Test(); t.test(); } void test() { Test: { test: for (int i = 0; true; i++) { if (i % 2 == 0) continue test; if (i > 10) break Test; System.out.print(i); } } }</code></p> <p>1) 1357 2) 13579 3) 135 4) 1 5) ошибка времени исполнения</p>
187.	<p>Дан следующий класс: <code>class abc.Point { abc.Point(int a, int b) { x = a; y = b; } int x, y; ... }</code> Укажите результат выполнения следующего фрагмента кода: <code>abc.Point p = new abc.Point(5,3), p1; p1 = p; p = null; System.out.println(p1.x);</code></p> <p>1) 3 2) 4 3) 5 4) будет выдано сообщение об ошибке 5) ничего из перечисленного</p>
188.	<p>Что будет выведено на экран в результате выполнения следующего кода: <code>public class SomeClass { static int x; public static void main(String[] args) { SomeClass s = new SomeClass(); s.test(5); System.out.println(x); } private void test(int x) { x = this.x + 5; System.out.print(x); } }</code></p> <p>1) 55 2) 510 3) 05 4) 50 5) сообщение об ошибке компиляции</p>
189.	<p>Что будет выведено на экран в результате выполнения следующего кода: <code>public</code></p>

	<pre>class SomeClass { public static void main(String s[]) { Calendar cal = new GregorianCalendar(); cal.set(Calendar.YEAR, 2011); cal.set(Calendar.MONTH, Calendar.APRIL); cal.set(Calendar.DAY_OF_MONTH, 30); cal.add(Calendar.DAY_OF_MONTH, 1); cal.roll(Calendar.MONTH, 11); SimpleDateFormat sdf = new SimpleDateFormat("yyyy MMMM dd"); System.out.println(sdf.format(cal.getTime())); } }</pre> <p>1) 2011 April 31 2) 2011 April 01 3) 2011 March 01 4) сообщение об ошибке компиляции 5) сообщение об ошибке времени исполнения</p>
190.	<p>Сколько комментариев в следующем примере кода: <code>int x = 0; /* text // text */ int y=1; // text */ // text */</code></p> <p>1) 1 блочный, 0 строчных 2) 1 блочный, 1 строчный 3) 1 блочный, 2 строчных 4) 2 блочных, 3 строчных</p>
191.	<p>Укажите количество преобразований, которые будут сделаны в следующем коде: <code>long a = 3; a = 5 + a; System.out.print("a = " + a);</code></p> <p>1) 1 2) 2 3) 3 4) 4 5) 5</p>
192.	<p>Приведенная ниже программа должна вывести на консоль Hello World! Выберите строки, которые нужно модифицировать в вашей программе, чтобы получить правильный результат. 1 <code>public class SomeClass {</code> 2 <code>public static void main(String[] args) {</code> 3 <code>String[] arr = {"H", "e", "l", "l", "o", " ", "W", "o", "r", "l", "d", "!"};</code> 4 <code>String result = "";</code> 5 <code>int i = 0;</code> 6 <code>...</code> 7 <code>System.out.print(arr[++i]);</code> 8 <code>}</code> 9 <code>}</code></p> <p>1) заменить строку 6 на <code>for(i = -1; i < arr.length;)</code> 2) заменить строку 6 на <code>for(i = -1; i < arr.length - 1;)</code> 3) заменить строку 6 на <code>for(i = -1; i < arr.length; i++)</code> 4) заменить строку 6 на <code>for(i = -1; i < arr.length - 1; i++)</code></p>
193.	<p>Укажите результат выполнения следующего фрагмента кода: <code>int a = 5; int b = 6; ++a = b++; System.out.println(a);</code></p> <p>1) 5 2) 6 3) 7 4) будет выдано сообщение об ошибке 5) ничего из перечисленного</p>
194.	<p>Какие варианты кода будут выполнены без ошибок?</p> <p>1) <code>int a = b; boolean c = (boolean)a;</code> 2) <code>byte b = 3; int a = b;</code> 3) <code>int a = 2; byte c = a;</code> 4) <code>int a = 2; float c = (float)a;</code></p>
195.	<p>Какой из следующих вариантов кода выведет на экран "perform"?</p> <p>1) <code>public class SomeClass { void perform() { System.out.println("perform"); } private SomeClass c; public static void main(String[] args) { SomeClass.perform(); } }</code> 2) <code>public class SomeClass { static void perform() { System.println("perform"); } private static SomeClass = new SomeClass(); public static void main(String[] args) { c.perform("perform"); } }</code> 3) <code>public class SomeClass { static void perform() { System.out.println("perform"); } private static SomeClass c; public static void main(String[] args) { SomeClass.c.perform(); } }</code> 4) <code>public class SomeClass { void perform() { System.out.println("perform"); } private static SomeClass c = new SomeClass(); public static void main(String[] args) { c.perform(); } }</code></p>
196.	<p>Что нужно подставить вместо знаков подчеркивания, чтобы программа вывела на экран "000"? <code>public class SomeClass { public static void main(String[] args) { _____ for (int i = 0; i < b.length; i++) { System.out.print(b[i]); } } }</code></p> <p>1) <code>int b[] = new {0,0,0};</code> 2) <code>int b[3] = [0,0,0];</code></p>

	<p>3) int b[] = {0,0,0}; 4) int b[] = [0,0,0];</p>
197.	<p>Есть следующий код: public class SomeClass { public static void main(String[] args) { SomeClass t = new SomeClass(); ____ } private int check(String x, int n) { if (n == 0) return n; else if (n == 1) { if (x != null) return 5; } else if (n == 2 && x != null) { if (x.equals("YES")) return 3; else if (x.equals("NO")) return 4; } return -1; } } Если вместо знаков подчеркивания поместить код вызова метода check, то какое из предложений будет верным?</p> <p>1) t.check("NO",2), в этом случае функция вернет значение 4 2) t.check("NO",1), в этом случае функция вернет значение 5 3) t.check("ANY",0), в этом случае функция вернет значение -1 4) t.check("YES",3), в этом случае функция вернет значение 3 5) t.check("YES",5), в этом случае функция вернет значение -1</p>
198.	<p>Дан следующий класс: class abc.Point { abc.Point(int a, int b) { x = a; y = b; } int x, y; ... } Укажите результат выполнения следующего фрагмента кода: abc.Point p = new abc.Point(5,3), p1; System.out.println(p1 == p);</p> <p>1) true 2) false 3) будет выдано сообщение об ошибке 4) ничего из перечисленного</p>
199.	<p>Какие варианты кода не вызовут ошибку компиляции?</p> <p>1) class Parent {} class FirstChild extends Parent {} class SecondChild implements FirstChild {} 2) class Parent {} interface FirstChild {} class SecondChild implements FirstChild {} 3) class Parent {} interface FirstChild extends Parent {} class SecondChild implements FirstChild {} 4) class Parent {} interface FirstChild {} interface SecondChild extends FirstChild {} 5) class Parent {} interface FirstChild {} interface SecondChild implements FirstChild {}</p>
200.	<p>Укажите результат выполнения следующего кода? long a = 111111111111L; float f = a; long b = 11111L; float d = b; System.out.println(a == (long) f); System.out.println(b == (long) d);</p> <p>1) true true 2) true false 3) false true 4) false false 5) будет выведено сообщение об ошибке</p>
201.	<p>Какой из следующих вариантов кода будет скомпилирован без ошибок</p> <p>1) public class SomeClass { static int a = SomeClass.b; static int b = a; } 2) public class SomeClass { static int a = b; int b = 5; } 3) public class SomeClass { static int a = SomeClass.b; int b = 5; } 4) public class SomeClass { static int a = SomeClass.b; static int b = 5; } 5) public class SomeClass { int a = SomeClass.b; static int b = 5; }</p>
202.	<p>Что будет выведено на экран в результате выполнения следующего кода: public class SomeClass { public static void main(String[] args) { SomeClass t = new SomeClass(); t.say(3); } private void say(int digit) { switch (digit) { case 1: System.out.print("ONE"); break; case 2: System.out.print("TWO"); case 3: System.out.print("THREE"); default: System.out.print("Unknown"); } } }</p> <p>1) ONETWO 2) TWO 3) TWOTHREEUnknown 4) THREEUnknown 5) ONETWOTHREEUnknown</p>
203.	<p>Что будет выведено на экран в результате выполнения следующего фрагмента кода: public class SomeClass { public static void main(String args[]) { SomeClass s = new SomeClass(); System.out.println(s.x + " " + s.y); } int getX() { return x; } int y = getX(); int x = 3; }</p> <p>1) 3, 3 2) 3, 0 3) 0, 0 4) сообщение об ошибке компиляции 5) ничего из перечисленного</p>
204.	<p>Что будет выведено на экран в результате выполнения следующего кода: public</p>

	<pre>class SomeClass { public void perform() { this.perform(1); System.out.print("1"); } public void perform(int val) { System.out.print(val); } public static void main(String[] args) { SomeClass c = new SomeClass2(); c.perform(4); } } class SomeClass2 extends SomeClass { public void perform() { super.perform(); System.out.print("3"); } public void perform(int val) { super.perform(); System.out.print(val); } }</pre> <p>(1) 114 (2) 4113 (3) 41 (4) 311 (5) произойдет ошибка времени выполнения</p>
205.	<p>Что будет выведено на экран в результате выполнения следующего кода: <pre>public class SomeClass { public static void main(String[] args) { int i = 0; do { if (i > 5) i--; if (i < 4) i++; if (i == 5) i+=2; i++; System.out.print(i); } while (i <= 5); } }</pre></p> <p>1) 2457 2) 24578 3) 2458 4) 245 5) ошибка времени выполнения</p>
206.	<p>Укажите возможный результат выполнения следующего кода: <pre>public abstract class SomeClass implements Runnable { public final static Object one = new Object(), two = new Object(); public static void main(String s[]) { Thread t1 = new Thread() { public void run() { synchronized (one) { synchronized (two) { System.out.print("1"); } } } }; Thread t2 = new Thread() { public void run() { synchronized (two) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } synchronized (one) { System.out.print("2"); } } } }; t1.start(); t2.start(); } }</pre></p> <p>1) программа не завершит работу 2) программа завершит работу, на консоли появится 1 3) программа завершит работу, на консоли появится 2 4) программа не завершит работу, на консоли появится 12 5) программа завершит работу, на консоли появится 12</p>
207.	<p>Укажите варианты кода, которые не вызовут ошибки компиляции.</p> <p>1) public void doSomething(int x, final int y) { x = Math.abs(y); } 2) <pre>public void doSomething(int x, final int y) { y = Math.abs(y); }</pre> 3) public void doSomething(int x, final int y) { x = Math.abs(y); return; } 4) <pre>public void doSomething(int x, final int y) { x = Math.abs(y); return x; }</pre> 5) <pre>public void doSomething(int x, final int y) { y = Math.abs(x); return; }</pre></p>
208.	<p>Укажите результат выполнения следующего кода? <pre>public class SomeClass { public static void main(String[] args) { SomeClass c = new SomeChild2(); ((SomeClass)c).print(); ((SomeChild1)c).print(); ((SomeChild2)c).print(); } public void print() { System.out.println("SomeClass"); } } class SomeChild1 extends SomeClass { public void print() { System.out.println("SomeChild1"); } } class SomeChild2 extends SomeClass { public void print() { System.out.println("SomeChild2"); } }</pre></p> <p>1) SomeClass SomeChild1 SomeChild2 2) SomeClass SomeClass SomeClass 3) SomeClass и сообщение об ошибке 4) SomeChild1 и сообщение об ошибке 5) SomeChild2 и сообщение об ошибке</p>
209.	<p>Какой метод сгенерирует ошибку <code>ArrayStoreException</code> при передаче в него массива <code>v</code>? <pre>public class Vehicle { public static void main(String[] args) { Vehicle[] v = new Car[] { new Car(); ... } public void setVehicles(Vehicle c[]) { c[0] = new Vehicle(); } public void setVehicles2(Vehicle c[]) { if (c[0] instanceof Vehicle) { c[0] = new Car(); } } public void setVehicles3(Vehicle c[]) { if (c[0] instanceof Car) { c[0] = new Vehicle(); } } } class Car extends Vehicle{ }</pre></p> <p>1) <code>setVehicles</code> 2) <code>setVehicles2</code> 3) <code>setVehicles3</code> 4) <code>setVehicles</code> и <code>setVehicles3</code> 5) <code>setVehicles</code> и <code>setVehicles2</code></p>
210.	<p>Есть следующая иерархия наследования: <code>java.lang.Object</code> <code>+-java.lang.Throwable</code> <code>+-java.lang.Exception</code> <code>+-java.lang.RuntimeException</code> <code>+++java.lang.IndexOutOfBoundsException</code> <code>+-java.lang.StringIndexOutOfBoundsException</code> <code>+-java.lang.ArrayIndexOutOfBoundsException</code> В методе <code>testSomeValue</code> могут быть возбуждены исключения <code>StringIndexOutOfBoundsException</code> и</p>

	<p>ArrayIndexOutOfBoundsException, при этом они не обрабатываются в блоке try – catch. Какие из перечисленных ниже утверждений будут верными?</p> <p>1) определение метода testSomeValue может включать throws StringIndexOutOfBoundsException, ArrayIndexOutOfBoundsException</p> <p>2) если метод, вызывающий testSomeValue, перехватывает IndexOutOfBoundsException, то исключение StringIndexOutOfBoundsException тоже будет перехватываться</p> <p>3) так как в определении метода указано throws StringIndexOutOfBoundsException, ArrayIndexOutOfBoundsException, любой вызывающий его метод должен перехватывать эти типы исключений, вне зависимости от того, возбуждается во время работы исключения или нет</p> <p>4) при объявлении метода testSomeValue надо обязательно указывать возбуждаемые исключения</p>
211.	<p>Каким будет результат работы следующего кода? public abstract class SomeClass implements Runnable{ public static Object lock = new Object(); public void lock() { synchronized (lock) { try { lock.wait(); System.out.print("1"); } catch (InterruptedException e) {} } } public void notifyThread() { synchronized (lock) { lock.notify(); } } public void unlock() { synchronized (lock) { lock.notify(); System.out.print("2"); } } public static void main(String s[]) { final SomeClass c = new SomeClass() { public void run() { lock(); notifyThread(); } }; final SomeClass c1 = new SomeClass() { public void run() { unlock(); } }; new Thread(c).start(); new Thread(c1).start(); } }</p> <p>1) программа не завершит работу, на консоли ничего не появится</p> <p>2) программа не завершит работу, на консоли появится 1</p> <p>3) программа не завершит работу, на консоли появится 2</p> <p>4) программа завершит работу, на консоли появится 12</p> <p>5) программа завершит работу, на консоли появится 21</p> <p>6) результат трудно предугадать</p>
212.	<p>Даны следующие классы: class abc.SomeClass {} class abc.SomeClass2 extends abc.SomeClass {} Что будет выведено на экран в следующем фрагменте кода: abc.SomeClass x1 = new abc.SomeClass(); abc.SomeClass2 x2 = new abc.SomeClass2(); System.out.println((x1 instanceof abc.SomeClass) + " " + (x2 instanceof abc.SomeClass2));</p> <p>1) false false</p> <p>2) true true</p> <p>3) true false</p> <p>4) false true</p>
213.	<p>Что будет выведено на экран в результате выполнения следующего кода? import java.awt.Point; public class Obscuring { static Point Test = new Point(3,2); public static void main (String s[]) { System.out.println(Test.x); } } class Test { static int x = -5; }</p> <p>1) -5</p> <p>2) 2</p> <p>3) 3</p> <p>4) будет выдано сообщение об ошибке</p>
214.	<p>Что будет выведено на экран в результате выполнения следующего кода? public class SomeClass { public static void main(String args[]) { SomeClass2 s = new SomeClass2(); } public SomeClass() { System.out.print("1"); } { System.out.print("2"); } } class SomeClass2 extends SomeClass { public SomeClass2() { System.out.print("3"); } { System.out.print("4"); } }</p> <p>1) 1243</p> <p>2) 2143</p> <p>3) 1234</p> <p>4) 2134</p> <p>5) 3421</p>
215.	<p>Укажите результат выполнения следующего кода: public class SomeClass { public static void main(String[] args) { SomeClass c = new SomeClass(); c.f1(1, 1); } public void f1(Integer i1, Integer i2) { System.out.println("1"); } public void f1(Integer i1, int i2) { System.out.println("2"); } public void f1(int b, Integer c) { System.out.println("3"); } public void f1(long b, long c) { System.out.println("4"); } }</p> <p>1) 1</p> <p>2) 2</p> <p>3) 3</p> <p>4) 4</p> <p>(5) возникнет ошибка компиляции</p>

216.	<p>Какие утверждения относительно клонирования верны?</p> <p>1) выражение <code>x.clone().getClass() == x.getClass()</code> должно быть истинным</p> <p>2) выражение <code>x == x.clone()</code> должно быть ложным</p> <p>3) попытка вызова <code>new Object().clone()</code> не приведет к ошибке</p> <p>4) попытка вызова <code>new Object().clone()</code> приведет к ошибке</p>
217.	<p>Предположим, нам необходимо создать собственную иерархию исключений. Рассмотрим следующий пример. <code>Exception +--LengthException +--TooLongException +--TooShortException</code> 1 class BaseMeasurer{ 2 int measureLength(Dimension d) throws TooShortException{..} 3 } 4 class DerivedMeasurer extends BaseMeasurer{ 5 ____ { 6 } 7 }</p> <p>Какое из перечисленных ниже выражений можно использовать в строке 5 вместо знаков подчеркивания, чтобы код успешно откомпилировался?</p> <p>1) <code>int measureLength(Dimension d) throws TooShortException</code></p> <p>2) <code>int measureLength(Dimension d) throws LengthException</code></p> <p>3) <code>int measureLength(Dimension d) throws TooLongException</code></p> <p>4) <code>int measureLength(Dimension d)</code></p>
218.	<p>Каким будет результат работы следующего кода? <code>public abstract class SomeClass implements Runnable{ private Object lock = new Object(); public void lock() { synchronized (lock) { try { lock.wait(); System.out.println("1"); } catch (InterruptedException e) {} } } public void notifyThread() { synchronized (lock) { lock.notify(); } } public void unlock() { synchronized (lock) { lock.notify(); System.out.println("2"); } } public static void main(String s[]) { Thread t = new Thread(new SomeClass() { public void run() { lock(); try { Thread.sleep(1000); } catch (InterruptedException e) {} notifyThread(); } }); t.start(); Thread t1 = new Thread(new SomeClass() { public void run() { unlock(); } }); t1.start(); } }</code></p> <p>1) программа не завершит работу, на консоли ничего не появится</p> <p>2) программа не завершит работу, на консоли появится 2</p> <p>3) программа не завершит работу, на консоли появится 1</p> <p>4) программа не завершит работу, на консоли появится 21</p> <p>5) программа не завершит работу, на консоли появится 12</p> <p>6) результат трудно предугадать</p>
219.	<p>Укажите результат выполнения следующего кода: <code>public class SomeClass { public static void main(String[] args) { SomeClass s = new SomeClass("5", "6"); SomeClass s1 = new SomeClass("10", "12"); s1.doSomething(s); System.out.println(s1.x + s1.y + s.x + s.y); } private void doSomething(SomeClass s) { s.x = "5"; this.y = "6"; } public SomeClass(String x, String y) { this.x = x; this.y = y; } String x, y; }</code></p> <p>1) 5656</p> <p>2) 101256</p> <p>3) 10656</p> <p>4) будет выведена ошибка компиляции</p> <p>5) ни один из перечисленных</p>
220.	<p>Какое значение появится на консоли после выполнения следующей программы? <code>class Parent { public void print() { System.out.println(x); } } class Child extends Parent { int x = 3; public static void main(String[] args) { Parent c = new Child(); c.print(); } }</code></p> <p>1) 1</p> <p>2) 2</p> <p>3) 3</p> <p>4) будет выведено сообщение об ошибке</p>
221.	<p>Что будет выведено на экран в результате выполнения следующего кода: <code>public class SomeClass { public static void main(String[] args) { int a[] = {1, 2, 3}; int b[] = (int[]) a.clone(); a[0] = 0; System.out.print(b[0]); } }</code></p> <p>1) 0</p> <p>2) 0, 2</p> <p>3) 1</p> <p>4) 1, 2</p> <p>5) сообщение об ошибке времени исполнения</p>
222.	<p>Что будет выведено на экран в результате выполнения следующего кода: <code>public class Test { public static void main(String[] args) { Test t = new Test(); t.test(); } void test() { Test:{ test: for (int i = 0; true; i++) { if (i % 2 == 0) continue Test; if (i > 10) break Test; System.out.print(i); } } }</code></p> <p>1) 12</p> <p>2) 123456789</p> <p>3) 13579</p> <p>4) 1357</p> <p>5) ошибка компиляции</p>

223.	<p>Дан следующий класс: <code>class abc.Point { abc.Point(int a, int b) { x = a; y = b; } int x, y; ... }</code> Каков будет результат следующего примера? <code>abc.Point p1 = new abc.Point(3, 4); abc.Point p2 = p1; p1.x = 5; p1 = new abc.Point(4, 4); System.out.print(p2.x - p1.x);</code></p> <p>1) -1 2) 0 3) 1 4) будет выдано сообщение об ошибке 5) ничего из перечисленного</p>
224.	<p>Какие варианты кода не приведут к ошибке компиляции?</p> <p>1) <code>public class SomeClass { public SomeClass(int x, int y) { this.x = x; this.y = y; super(); } int x, y; }</code> 2) <code>public class SomeClass { public SomeClass(int x, int y) { super(); this.x = x; this.y = y; } int x, y; }</code> 3) <code>public class SomeClass { public SomeClass(int x, int y) { super(); this.x = x; this.y = y; } public SomeClass() { super(); this(0, 0); } int x, y; }</code> 4) <code>public class SomeClass { public SomeClass(int x, int y) { super(); this.x = x; this.y = y; } public SomeClass() { this(0, 0); } int x, y; }</code> 5) <code>public class SomeClass { public SomeClass(int x, int y) { this(x, y); } int x, y; }</code></p>
225.	<p>Какие из следующих присвоений корректны?</p> <p>1) <code>byte c = (char)150;</code> 2) <code>byte c = (char)50;</code> 3) <code>byte c = 'a';</code> 4) <code>byte c = (char)150 + "a";</code> 5) <code>byte c = (long)50;</code></p>
226.	<p>Укажите корректный вариант кода</p> <p>1) <code>interface SomeInterface{} class SomeClass extends SomeInterface{ }</code> 2) <code>interface SomeInterface{} interface SomeInterface2 extends SomeInterface{} class SomeClass implements SomeInterface2{ }</code> 3) <code>interface SomeInterface{} interface SomeInterface2 extends SomeInterface{} class SomeClass implements SomeInterface2 extends SomeInterface { }</code> 4) <code>interface SomeInterface{} interface SomeInterface2 extends SomeInterface{} class SomeClass implements SomeInterface2, SomeInterface { }</code> 5) <code>interface SomeInterface{} interface SomeInterface2 extends SomeInterface{} class SomeClass implements SomeInterface extends SomeInterface2 { }</code></p>
227.	<p>Есть следующий код: <code>public class SomeClass { public static void main(String[] args) { SomeClass t = new SomeClass(); ____ } private int check(String x, int n) { if (n == 0) return n; else if (n == 1) { if (x != null) return 5; } else if (n == 2 && x != null) { if (x.equals("YES")) return 3; else if (x.equals("NO")) return 4; } return -1; } }</code> Если вместо знаков подчеркивания поместить код вызова метода <code>check</code>, то какое из предложений будет верным?</p> <p>1) <code>t.check("ANY",1)</code>, в этом случае функция вернет значение 5 2) <code>t.check("NO",2)</code>, в этом случае функция вернет значение 3 3) <code>t.check("YES",1)</code>, в этом случае функция вернет значение -1 4) <code>t.check("YES",2)</code>, в этом случае функция вернет значение 3 5) <code>t.check("ANY",6)</code>, в этом случае функция вернет значение -1</p>
228.	<p>Дан следующий класс: <code>class abc.Point { abc.Point(int a, int b) { x = a; y = b; } int x, y; ... }</code> Укажите результат выполнения следующего фрагмента кода: <code>abc.Point p = new abc.Point(5,3), p1 = p; System.out.println(p1 == p);</code></p> <p>1) true 2) false 3) будет выдано сообщение об ошибке 4) ничего из перечисленного</p>
229.	<p>Укажите результат выполнения следующего кода? <code>long a = 111111111111L; double f = a; long b = 11111L; double d = b; System.out.println(a == (long) f); System.out.println(b == (long) d);</code></p> <p>1) true true 2) true false 3) false true 4) false false 5) будет выведено сообщение об ошибке</p>
230.	<p>Какой из следующих вариантов кода будет скомпилирован без ошибок</p> <p>1) <code>public class SomeClass { static int a = 5; static int b = a; }</code></p>

	<p>2) <code>public class SomeClass { static int a = b; int b = 5; }</code> 3) <code>public class SomeClass { static int a = SomeClass.b; int b = 5; }</code> 4) <code>public class SomeClass { static int a = SomeClass.b; static int b = 5; }</code> 5) <code>public class SomeClass { int a = SomeClass.b; static int b = 5; }</code></p>
231.	<p>Какие преобразования не вызовут ошибку компиляции? 1) <code>Object o = (String)null; String s = o;</code> 2) <code>Object o = (String)null; String s = (String)o;</code> 3) <code>Object o = null; String s = (String)o;</code> 4) <code>Object o = null; String s = o;</code></p>
232.	<p>Что нужно подставить вместо знаков подчеркивания, чтобы программа вывела на экран "12": <code>public class SomeClass { public void perform() { System.out.print("1"); } public static void main(String[] args) { SomeClass c = new SomeClass2(); c.perform(); } } class SomeClass2 extends SomeClass { public void perform() { ____ System.out.print("2"); } }</code> 1) <code>this.perform();</code> 2) <code>super.perform();</code> 3) <code>perform();</code> 4) <code>SomeClass.perform();</code> 5) ничего подставлять не надо</p>
233.	<p>Укажите возможный результат выполнения следующего кода: <code>public abstract class SomeClass implements Runnable { public final static Object one = new Object(), two = new Object(); public static void main(String s[]) { Thread t1 = new Thread() { public void run() { synchronized (one) { synchronized (two) { System.out.print("1"); } } } }; Thread t2 = new Thread() { public void run() { synchronized (two) { synchronized (one) { System.out.print("2"); } } } }; t1.start(); t2.start(); } }</code> 1) программа не завершит работу, на консоли появится 1 2) программа не завершит работу, на консоли появится 2 3) программа не завершит работу, на консоли появится 12 4) программа завершит работу, на консоли появится 12</p>
234.	<p>Какой тип должна иметь переменная d в следующем фрагменте: <code>int a = 1; short b = 1; long c = 1; c = (d = a + c) ? a : b;</code> 1) <code>char</code> 2) <code>byte</code> 3) <code>long</code> 4) <code>int</code> 5) <code>short</code></p>
235.	<p>Укажите результат выполнения следующего кода? <code>public class SomeClass { public static void main(String[] args) { SomeClass c = new SomeChild1(); ((SomeClass)c).print(); ((SomeChild1)c).print(); ((SomeChild2)c).print(); } public void print() { System.out.println("SomeClass"); } } class SomeChild1 extends SomeClass { public void print() { System.out.println("SomeChild1"); } } class SomeChild2 extends SomeClass { public void print() { System.out.println("SomeChild2"); } }</code> 1) <code>SomeClass SomeChild1 SomeChild2</code> 2) <code>SomeClass SomeClass SomeClass</code> 3) <code>SomeClass</code> и сообщение об ошибке 4) <code>SomeChild1 SomeChild1</code> и сообщение об ошибке 5) <code>SomeClass SomeClass</code> и сообщение об ошибке</p>
236.	<p>Какой метод сгенерирует ошибку <code>ArrayStoreException</code> при передаче в него массива v? <code>public class Vehicle { public static void main(String[] args) { Vehicle[] v = new Car[] { new Car(); ... } public void setVehicles(Vehicle c[]) { c[0] = new Vehicle(); } public void setVehicles2(Vehicle c[]) { if (c[0] instanceof Car) { c[0] = new Vehicle(); } } public void setVehicles3(Vehicle c[]) { if (c[0] instanceof Vehicle) { c[0] = new Car(); } } } class Car extends Vehicle{ }</code> 1) <code>setVehicles</code> 2) <code>setVehicles2</code> 3) <code>setVehicles3</code> 4) <code>setVehicles</code> и <code>setVehicles3</code> 5) <code>setVehicles</code> и <code>setVehicles2</code></p>
237.	<p>Что будет выведено на экран в результате выполнения следующего кода: <code>public class Test { public static void main(String[] args) { Test t = new Test(); t.test(); } void test() { Test: { test: for (int i = 0; true; i++) { if (i % 2 == 0) continue test; if (i > 10) break test; System.out.print(i); } } }</code> 1) 12</p>

2) 123456789
3) 13579
4) 1357
5) ошибка времени исполнения

3.3 Вопросы к экзамену

3.4.1 УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

№	Текст вопроса
238.	Основные достоинства языка Java.
239.	Основные недостатки. Основные особенности.
240.	Версии Java.
241.	Апплеты.
242.	Жизненный цикл программы на Java.
243.	Структура пакета SDK.
244.	ООП в Java, наследование.
245.	Объектноориентированный язык программирования.
246.	5 принципов объектноориентированного подхода.
247.	Реализация принципов объектно-ориентированного подхода в Java.
248.	Операции в Java.
249.	Операции (operators) в языке Java.
250.	Литералы (константы).
251.	Арифметические. Логические литералы.
252.	Строковые литералы.
253.	Символьные литералы.
254.	Операторы (statements).
255.	Оператор - выражение.
256.	Условный оператор (if).
257.	Оператор return.
258.	Оператор цикла по предусловию (while).
259.	Оператор цикла по постусловию (do while).
260.	Оператор цикла "со счетчиком" (for).
261.	Операторы break и continue.
262.	Оператор выбора (switch).

ПКв-8 Способность проводить тестирование компонентов программного обеспечения ИС.

№	Текст вопроса
263.	Массивы в Java.
264.	Массивы объектов.
265.	Многомерные массивы.
266.	Присваивание и копирование.
267.	Резюмируем основные правила.
268.	Конструкторы классов.
269.	Вызов одного конструктора из другого.
270.	Работа со строками (класс String).
271.	Структура Java программ.
272.	Знакомство с библиотеками и пакетами.

ПКв-9 Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

№	Текст вопроса
273.	Использование пакетов в программах.
274.	Создание своих собственных пакетов.
275.	Коллекции элементов.
276.	Статические вложенные классы.

277.	Коллекции объектов Java.
278.	Коллекции - это наборы произвольных объектов.
279.	Итераторы. Классы реализации коллекций.
280.	Коллекции- списки (List).
281.	Исключительные ситуации.
282.	Обработка исключительных ситуаций (Exceptions).
283.	Поведение программы при возникновении исключения.
284.	Структура и использование блока перехвата исключений.
285.	Классы исключительных ситуаций. Механизм контроля перехвата исключений.

3.4 Экзамен (примеры заданий)

ПКв-9 Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

№ задания	примеры заданий
286.	<p>Написать программу, получающую на вход в качестве аргумента два параметра - числа x и y. Если сумма этих чисел больше 20, то вычислить утроенный квадрат первого числа, в противном случае куб второго числа. Вывести результат на экран.</p> <pre> package pkg1; import static java.lang.Math.*; public class Main { public static void main(String[] args) { double x = Double.parseDouble(args[0]); double y = Double.parseDouble(args[1]); System.out.println(" первое число = " + x); System.out.println("второе число = " + y); if (x + y > 20) { System.out.println ("первое число, утроенный квадрат = "+ x*18); } else { System.out.println ("второе число, квадрат =" + y* y); } } } </pre>
287.	<p>Дано a,b,. Найти наибольшее max и наименьшее min . Найти d:</p> $d = \frac{\max(2 \cdot a, b) - 10 \cdot \sqrt{\min(a, 4 + b)} + 4.2 \min(a, 4 + b)}{1 + \frac{\max(2a, b)}{\min(a, 4 + b)}}$ <pre> package pkg1; import static java.lang.Math.*; public class Main { public static void main(String[] args) { double a = Double.parseDouble(args[0]); double b = Double.parseDouble(args[1]); double d; System.out.println(" a = " + a); System.out.println("b = " + b); if (a>b) { d = (2*a-10*sqrt (4+b)+4.2*4+b)/ (1 + 2*a/4+b); System.out.println ("d = " + d); } else { d = (2*b-10*sqrt (a)+4.2*a) / (1 + b/a); System.out.println ("d = " + d); } } </pre>

	<pre> } } </pre>
288.	<p>Написать программу, получающую на вход в качестве аргумента два параметра - числа x и y. Большее из них возвести в квадрат, из меньшего извлечь корень квадратный. Вывести результат на экран.</p> <p style="text-align: center;">Листинг программы</p> <pre> package practic_work_1; public class Task_1 { public static void ComparisonAndCalculation(double first_num, double second_num) { double buff_first_num = first_num, buff_second_num = second_num; if (first_num > second_num) { first_num = first_num * first_num; second_num = Math.sqrt(second_num); System.out.println("Наименьшее число - args[1] = " + buff_second_num + " стало равным " + second_num); System.out.println("Наибольшее число - args[0] = " + buff_first_num + " стало равным " + first_num); } else { second_num = second_num * second_num; first_num = Math.sqrt(first_num); System.out.println("Наибольшее число - args[1] = " + buff_second_num + " стало равным " + second_num); System.out.println("Наименьшее число - args[0] = " + buff_first_num + " стало равным " + first_num); } } public static void main(String[] args) { double first_num = Double.parseDouble(args[0]), second_num = Double.parseDouble(args[1]); ComparisonAndCalculation(first_num, second_num); } } </pre>
289.	<p>5. Дано a,b. Найти наибольшее max(a,b) из чисел a и b. Найти $d:=tg(\max(a,b))$</p> <p style="text-align: center;">Листинг программы</p> <pre> package practic_work_1; import static java.lang.Math.*; public class Task_2 { public static void main(String[] args) { double first_num = Double.parseDouble(args[0]), second_num = Double.parseDouble(args[1]); System.out.println("Максимальное число: " + max(first_num,second_num)); } } </pre>

```

        System.out.println("Минимальное число: " +
min(first_num,second_num));
        double d = Math.tan(max(first_num,second_num));
        System.out.println("d = " + d);
    }
}}

```

290.

Вариант № 23

23. Наиболее рационально вычислить Y и F. Использовать составной оператор.

$$Y = \begin{cases} x^3 + 1 & \text{если } x \leq -3 \\ (1 + 2^{\lg x}) & \text{если } -3 < x \leq 0 \\ \operatorname{ctg}^2 x & \text{если } x > 0 \end{cases} \quad F = \begin{cases} e^{x+1} & \text{если } x \leq -3 \\ x^4 & \text{если } -3 < x \leq 0 \\ \sqrt[5]{\lg x} & \text{если } x > 0 \end{cases}$$

Листинг программы

```

package practic_work_1;
import static java.lang.Math.*;

public class Task_3 {
    public static double CalculationY(double x) {
        double Y = 0;
        if (x <= -3)
        {
            Y += pow(x,3)+1;
        }
        else if (x > 0)
        {
            Y += pow((1.0/Math.tan(x)),2);
        }
        else
        {
            Y += 1+pow(2,tan(x));
        }
        return Y;
    }
    public static double CalculationF(double x) {
        double F = 0;
        double EPSILON = 1.0E-6;
        if (x <= -3)
        {
            F += pow(EPSILON,x+1);
        }
        else if (x > 0)
        {
            F += pow(x,4);
        }
        else
        {
            F += pow(Math.tan(x),1/5);
        }
        return F;
    }
    public static void main(String[] args) {
        double x = Double.parseDouble(args[0]);
        double ans_Y = CalculationY(x);
        double ans_F = CalculationF(x);
        System.out.println("При x = " + x);
    }
}

```

	<pre> System.out.println("Y = " + ans_Y); System.out.println("F = " + ans_F); } } </pre>
291.	<p>Написать программу, получающую на вход в качестве аргумента два параметра - числа a и b. Если произведение двух чисел больше 30, то вычислить удвоенный котангенс первого числа, в противном случае первое число разделить на 2. Вывести результат на экран.</p> <pre> import static java.lang.Math.*; public class HelloWorld { public static void main(String [] args) { int mult; int a = 2; int b = 10; mult = a*b; if (mult > 30) { System.out.println(2*(cos(a)/sin(a))); } else { System.out.println(a/2) ; }}} </pre>
292.	<p>11. Дано a,b,. Найти наибольшее max и наименьшее min . Найти d:</p> $d = \frac{\min(a, 3 \cdot b) - 2 \cdot \sqrt{\max(a, 3 \cdot b)}}{4^3 + \frac{\max(a, 3 \cdot b)}{\min(a, b)}}$ <pre> import static java.lang.Math.*; public class HelloWorld { public static void main(String [] args) { int min; int max; int a = 9; int b = 5; double d; if (a > b) { max = a; } else { max = b; } if (a < b) { min = a; } else { min = b; } d = ((min(a,3*b)- 2*sqrt(max(a,3*b)))/(pow(4,3)+((max(a,3*b))/(min(a,b))))); System.out.println("d = " + d); }} </pre>
293.	Задание 3.

11. Наиболее рационально вычислить Y и F. Использовать составной оператор.

$$Y = \begin{cases} x^3 + \arctg 8x & \text{если } x \leq 1 \\ 5x - (1 + 3^x) & \text{если } 1 < x \leq 2 \\ \text{ctg}(2x+1) & \text{если } x > 2 \end{cases} \quad F = \begin{cases} e^{\text{tg} x + 1} & \text{если } x \leq 1 \\ x^4 - \text{tg} 4x & \text{если } 1 < x \leq 2 \\ \sqrt{x} & \text{если } x > 2 \end{cases}$$

```
import static java.lang.Math.*;
public class HelloWorld {
public static void main( String [] args) {

    double Y = 0;
    double F = 0;
    double x = -2;

    if (x <= 1) {
Y = pow(x,5)+ atan(8*x); }
    if (x > 1 && x <= 2) {
        Y = 5*x-(1+pow(3,x)); }
    if (x > 2) {
        Y = (cos(x)/sin(x))*(2*x+1); }

    if (x <= 1) {
        F = pow(exp(1),(tan(x)+1)); }
    if (x > 1 && x <= 2) {
        F = pow(x,4)-tan(4*x); }
    if (x > 2) {
        F = pow((sqrt(x)),(1/5)); }

    System.out.println("Y = " + Y);
    System.out.println("F = " + F); }}
```

294. Написать программу, получающую на вход в качестве аргумента два параметра - числа а и b. Если произведение двух чисел больше 20, то вычислить котангенс второго числа, в противном случае первое число разделить на 3. Вывести результат на экран.

Исходный код:

```
public static void firstEx() {
    Scanner in = new Scanner(System.in);
    System.out.print("Введите a: ");
    double a = in.nextDouble();
    System.out.print("Введите b: ");
    double b = in.nextDouble();
    if (a * b > 20)
        System.out.println(1 / tan(b));
    else
        System.out.println(a / 3);
}
```

295. Дано a,b. Найти наибольшее max(a,b) и наименьшее min(a,b) из чисел а и b. Найти d:

$$d = \frac{\min(a,b) - 2 \cdot \sqrt{\max(a,b)}}{1 + \frac{\max(a,b)}{\min(a,b)}}$$

```
public static void secondEx() {
    Scanner in = new Scanner(System.in);
    System.out.print("Введите a: ");
    double a = in.nextDouble();
    System.out.print("Введите b: ");
    double b = in.nextDouble();
```

	<pre>System.out.println(max(a, b)); System.out.println(min(a, b)); System.out.println((min(a, b) - 2 * sqrt(max(a, b))) / (1 + max(a, b) / min(a, b))); }</pre>
296.	<p>Наиболее рационально вычислить Y и F. Использовать составной оператор.</p> $Y = \begin{cases} x^5 - \operatorname{tg}(2x - 1) & \text{если } x \leq -2 \\ 3x(1 + e^{x+1}) & \text{если } -2 < x \leq 1 \\ \operatorname{Sin}^5 x & \text{если } x > 1 \end{cases} \quad F = \begin{cases} e^{\sin x} & \text{если } x \leq -2 \\ x^2 & \text{если } -2 < x \leq 1 \\ \sqrt{\cos x} & \text{если } x > 1 \end{cases}$ <p>Рис. 2 – принимаемые значения Y и F</p> <p>Исходный код:</p> <pre>public static void thirdEx() { System.out.print("Введите x: "); Scanner in = new Scanner(System.in); double x = in.nextDouble(); double F = 0.0; double Y = 0.0; if (x <= -2) { F = pow(E, sin(x)); } else if (-2 < x && x <= 1) { F = pow(x, 2); } else { F = sqrt(cos(x)); } if (x <= -2) { Y = pow(x, 5) - tan(2 * x - 1); } else if (-2 < x && x <= 1) { Y = 3 * x * (1 + pow(E, x + 1)); } else { Y = pow(sin(x), 5); } System.out.println("Y = " + Y + "\nF = " + F); }</pre>
297.	<p>Написать программу, получающую на вход в качестве аргумента два параметра - числа x и y. Большее из них умножить на 2, меньшее разделить на 3. Вывести результат на экран.</p> <p>Листинг программы</p> <pre>class Main { static void getNumbers(double x, double y) { System.out.println("Исходные числа:\nx = " + x + "\ny = " + y + "\n"); if (x > y) { x *= 2; y /= 3; } else { x /= 3; y *= 2; } System.out.println("Результат:\nx = " + x + "\ny = " + y); } public static void main(String[] args) { System.out.println("Задание 1. Вариант 22\n"); } }</pre>

```

        getNumbers(14.2, 28.3);
    }
}

```

298.

4. Дано a,b,. Найти наибольшее **max** и наименьшее **min** . Найти d:

$$d = \frac{\max(2 \cdot a, b) - 10 \cdot \sqrt{\min(a, 4 + b)} + 4.2 \min(a, 4 + b)}{1 + \frac{\max(2 \cdot a, b)}{\min(a, 4 + b)}}$$

Листинг программы

```

import java.lang.Math.*;
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        System.out.println("Задание 2. Вариант 4\n");

        Scanner sc = new Scanner(System.in);

        System.out.println("Введите 2 числа:");
        System.out.print("a = ");
        int a = sc.nextInt();
        System.out.print("b = ");
        int b = sc.nextInt();

        double d = ( Math.max(2*a,b) -
10*Math.sqrt(Math.min(a,4+b))+4.2*Math.min(a,4+b) )
                / ( 1 + ( Math.max(2*a,b) ) / (Math.min(a,4+b)) );

        System.out.println("\nМаксимальное число max = " +
Math.max(2*a,b));

        System.out.println("Минимальное число min = " + Math.min(a,4+b));

        System.out.println("\nd= " + d);
    }
}

```

22. Наиболее рационально вычислить Y и F. Использовать составной оператор.

$$Y = \begin{cases} x^5 - \operatorname{tg}(2x - 1) & \text{если } x \leq -2 \\ 3x(1 + e^{x+1}) & \text{если } -2 < x \leq 1 \\ \operatorname{Sin}^5 x & \text{если } x > 1 \end{cases} \quad F = \begin{cases} e^{\sin x} & \text{если } x \leq -2 \\ x^2 & \text{если } -2 < x \leq 1 \\ \sqrt{\cos x} & \text{если } x > 1 \end{cases}$$

Листинг программы

```
import java.lang.Math.*;
import java.util.Scanner;

public class Main {

    static double CalculationOfY(int x) {

        double Y = 0;

        if (x <= -2) {

            Y += (Math.pow(x, 5) - Math.tan(2 * x - 1));

        } else if (x > 1) {

            Y += (3 * x * (1 + Math.exp(x + 1)));

        } else {

            Y += (Math.pow(Math.sin(x), 5));

        }

        return Y;

    }

    static double CalculationOfF(int x) {

        double F = 0;

        if (x <= -2) {

            F += (Math.exp(Math.sin(x)));

        } else if (x > 1) {

            F += x*x;

        } else {

            F += Math.sqrt(Math.cos(x));

        }

        return F;

    }

}
```

	<pre> public static void main(String[] args) { System.out.println("Задание 3. Вариант 22\n"); Scanner sc = new Scanner(System.in); System.out.print("Введите x: "); int x = sc.nextInt(); double Y = CalculationOfY(x); double F = CalculationOfF(x); System.out.println("Y = " + Y); System.out.println("F = " + F); } } </pre>
<p>300.</p>	<p>Написать программу, получающую на вход в качестве аргумента два параметра - числа x и y. Если произведение этих чисел больше 50, то вычислить удвоенный корень квадратный первого числа. Вывести результат на экран.</p> <p>Решение:</p> <pre> package javatest; import static java.lang.Math.*; import java.util.Scanner; /** * * @author User */ public class JavaTest { /** * @param args the command line arguments */ public static void main(String[] args) { // TODO code application logic here Scanner s=new Scanner(System.in); int x=s.nextInt(); Scanner l=new Scanner(System.in); int y=s.nextInt(); int mult = x * y; if(x*y>50) System.out.println(sqrt(x)); } } </pre>
<p>301.</p>	<p>Дано a,b. Найти наибольшее max(a,b) из чисел a и b. Найти d:=tg(max(a,b))</p> <p>Решение:</p> <pre> package javatest2; </pre>

	<pre> import static java.lang.Math.*; /** * * @author User */ public class Javatest2 { /** * @param args the command line arguments */ public static void main(String[] args) { // TODO code application logic here double a = 0; double b = 4; double j=max(a,b); double d=tan(j); System.out.println(d); } } </pre>
302.	<p>Наиболее рационально вычислить Y и F. Использовать составной оператор. $1 - x^5 + \tan 8x$ если $x \leq 0$ и $2x+1$ если $x \leq 0$ $Y = (1 + 3x)$ если $0 < x \leq 3$ $F = x^2 - \sin 4x$ если $0 < x \leq 3$ $\arctg(2x+1)$ если $x > 3$ $5/2$ x если $x > 3$</p> <p>Решение:</p> <pre> package javatest2; import static java.lang.Math.*; import java.util.Scanner; /** * * @author User */ public class Javatest2 { /** * @param args the command line arguments */ public static void main(String[] args) { // TODO code application logic here Scanner s=new Scanner(System.in); double x=s.nextDouble(); double Y=0.0; double F=0.0; if(x<=0) { Y=(1-pow(x,5)+tan(8*x)); F=(pow(E,2*x+1)); System.out.println("Y="+Y); System.out.println("F="+F); } else if(x<=0 x>0) { Y=(1+pow(3,x)); F=(pow(x,2)-sin(4*x)); System.out.println("Y="+Y); System.out.println("F="+F); } else if(x>3) { Y=(atan(2*x+1)); F=(pow(x,2/5)); System.out.println("Y="+Y); System.out.println("F="+F); } } } </pre>

	<pre> } } </pre>
303.	<p>Определить дополнительный массив разрешенных значений. Определить и вывести на экран, сколько элементов исходного массива имеют разрешенные значения;</p> <pre> package pkg1; import static java.lang.Math.*; public class Main { public static void main(String[] args) { int size1 = 20; int size2 = 7; int[] b = new int[size1]; int[] c = new int[size2]; for (int i = 0; i < size1; i++) { b [i] = ((int) ((Math.random() - 0.5) * 100)); } for (int i = 0; i < size2; i++) { c [i] = ((int) ((Math.random() - 0.5) * 100)); } int count=0; for (int i = 0; i < size1; i++) for (int j = 0; j < size2; j++) { if (b[i]== c[j]) count ++; } printVector (b) ; printVector (c) ; System.out.println(count); } private static void printVector(int[] vector) { for (int i = 0; i < vector.length; i++) { System.out.print(vector[i] + "\t");} System.out.println(); } } </pre>
304.	<p>Дан массив b (n) . Переписать в массив C(n) отрицательные элементы массива b(n) умноженные на 4. (со сжатием., без пустых элементов внутри). Затем упорядочить методом «выбора и перестановки» по возрастанию новый массив.</p> <pre> package pkg1; import static java.lang.Math.*; public class Main { public static void main(String[] args) { int N = 10; int[] b = new int[N]; int[] c = new int[N]; for (int i = 0; i < N; i++) { b [i] = ((int) ((Math.random() - 0.5) * 100)); } int j =0; for (int i = 0; i < N; i++) { if (b [i] < 0) { c [j] = b [i]*4; j++; } } } </pre>

	<pre> } sort (c); printVector (b) ; printVector (c) ; } public static void sort(int[] array) { for (int i = 0; i < array.length - 1; ++i) { int minPos = i; for (int j = i + 1; j < array.length; ++j) { if (array[j] < array[minPos]) { minPos = j; } } int saveValue = array[minPos]; array[minPos] = array[i]; array[i] = saveValue; } } private static void printVector(int[] vector) { for (int i = 0; i < vector.length; i++) { System.out.print(vector[i] + "\t");} System.out.println(); } } </pre>
305.	<p>максимальное значение в i-м столбце;</p> <pre> package pkg1; import static java.lang.Math.*; public class Main { public static void main(String[] args) { int[][] a=new int[5][5]; int max,min ; int [] vector= new int [5]; for (int i=0;i < a.length;i++){ for (int j=0;j < a[i].length;j++){ a[i][j]=(int)(Math.random()*10); } } for (int i=0;i < a.length;i++,System.out.println()){ for (int j=0;j < a[i].length;j++){ System.out.print(a[i][j]+" "); } } System.out.println (); Print Vector (vector, a); } private static void print Vector (int[] vector, int[][] a) { for (int j = 0; j < 5; j++) { vector [j] = a[0][j]; for (int i = 1; i < 5; i++) if (vector[j] < a[i][j]) vector[j] = a[i][j]; } for (int i=0;i < vector.length;i++){ System.out.print(vector[i]+" "); } } } } </pre>
306.	<p>Дан двумерный массив А, размером (n×n) (или квадратная матрица А). Найти произведение минимального и максимального элементов её главной диагонали. Затем умножить побочную диагональ на</p>

максимальный элемент главной диагонали.

```
package pkg1;
import static java.lang.Math.*;
public class Main {
    public static void main(String[] args) {

        int[][] a=new int[5][5];
        for (int i=0;i < a.length;i++){
            for (int j=0;j < a[i].length;j++){
                a[i][j]=(int)(Math.random() - 0.5) * 10);
            }
        }
        printMatrix(a);
        System.out.println();
        int max = searchMax(a);
        int min = searchmin (a);

        for (int i = a.length - 1; i >= 0; i--) {
            a[i][a.length - 1 - i] *= max;
        }
        System.out.println();
        printMatrix(a);
    }

    private static int searchMax (int[][] a)
    {
        int max = Integer.MIN_VALUE;
        for(int i = 0; i < a.length; i++){
            if (a[i][i] > max) max = a[i][i];
        }
        System.out.println("max = " + max);
        return max;
    }

    private static int searchmin (int[][] a)
    {
        int min = Integer.MAX_VALUE;
        for(int i = 0; i < a.length; i++){
            if (a[i][i] < min) min = a[i][i];
        }
        System.out.println("min = " + min);
        return min;
    }

    public static void printMatrix(int[][] a) {
        for (int i=0;i < a.length;i++,System.out.println()){
            for (int j=0;j < a[i].length;j++){
                System.out.print(a[i][j] + "\t");
            }
        }
    }
}
```

307.) Составить и вывести на экран массив номеров элементов исходного массива, встречающихся один раз;

```

package ns2;
import static java.lang.Math.*;
public class ns22 {
    public static void main(String[] args){
        int A[]=new int[10];
        int B[]=new int[10];
        int q=0;
        int count;
        System.out.print("Исходный массив:"+"\n");
        for (int i=0;i<10;i++)
        {
            A[i]=(int)Math.round(Math.random()*10);
            System.out.print(A[i]+" ");
        }
        System.out.print("\nНовый массив:");
        for (int i=0;i<10;i++)
        {
            count=0;
            for (int j=1;j<10;j++)
            {
                if(A[i]==A[j] && i!=j)
                    count++;
            }
            if(count==0)
            {
                B[q]=A[i];
                q++;
            }
        }
        for (int i=0;i<q;i++)
        {
            System.out.print(B[i]+" ");
        }
    }
}

```

ns2.ns22 >> main >> for (int i = 0; i < 10; i++) >>

рд ×

ns2 (debug) × Консоль отладки ×

```

debug:
Исходный массив:
10 2 6 9 6 8 6 9 1 4
Новый массив:10 2 8 1 4 СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 сек)

```

308. Дан массив с (n) . Переписать в массив х (n) все ненулевые элементы массива умноженные на 4. (со сжатием., без пустых элементов внутри). Затем упорядочить методом «выбора и перестановки»по возрастанию новый массив.

```

package ns2;
public class Ps2
{
    public static void main(String[] args)
    {
        int tm;
        int tp;
        int tmp;
        int q = 0;
        int b []= new int [5];
        int C []= new int [5];
        System.out.print("Вектор c(n)+"\n");
        for (int i=0; i < 5;i++)
        {
            b[i] = (int) Math.round(Math.random()*5);
            System.out.print(b[i] + " ");
        }
        System.out.println();
        for (int i=0; i < 5;i++)
        {
            if (b[i] != 0)
            {
                C[q] = b[i]*4;
                q++;
            }
        }

        System.out.print("Вектор x(n)+"\n");
        for (int i=0; i < q;i++)
        {
            System.out.print(C[i] + " ");
        }
        System.out.print("\nОтсортированный вектор x(n)+"\n");
        for (int i=0; i < q;i++)
        {
            tm = C[i];
            tp = i;
            for (int j=i+1; j < q;j++)
            {
                if (C[j] < tm)
                {
                    tm = C[j];
                    tp = j;
                }
            }
            if (tp != i)
            {
                tmp = C[i];
                C[i] = C[tp];
                C[tp] = tmp;
            }
        }
        for (int i=0; i < q;i++)
        {
            System.out.print(C[i] + " ");
        }
    }
}

```

ns2.Ps2 > main > tmp >

✕

ns2 (debug) ✕ Консоль отладки ✕

```

Вектор c(n)
1 4 1 3 1
Вектор x(n)
4 16 4 12 4
Отсортированный вектор x(n)
4 4 4 12 16 СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

```

309.) Определить матрицу (двумерный массив) и ее заполнить случайными значениями. Построить вектор В, которой возвращает – равно 1, если значения элементов i-й строки упорядочены по возрастанию, и 0, в противном случае;

```

package ns2;
import static java.lang.Math.*;
public class ns23
{
    public static void main(String[] args)
    {
        int n=3;
        int a[][]=new int [n][n];
        int b[]=new int [n];
        int k=0;
        System.out.print ("Матрица:"+"\n");
        for (int i=0;i<n;i++)
        {
            for (int j=0;j<n;j++)
            {
                a[i][j]=(int)Math.round(Math.random()*10);
                System.out.print(a[i][j]+" ");
            }
            System.out.println();
        }

        for (int i=0;i<n;i++)
        {
            for (int j=0;j<n-1;j++)
            {
                if (a[i][j]>=a[i][j+1])
                {
                    k=1;
                }
                else
                {
                    k=0;
                }
            }
            b[i]=k;
            System.out.print(b[k]);
        }
    }
}

```

ns2.ns23 > main >

од ×

ns2 (debug) × Консоль отладки ×

```

debug:
Матрица:
8 7 3
6 2 0
9 5 0
001111СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

```

310. Дан двумерный массив A, размером (n×n) (или квадратная матрица A). Найти минимальный элемент среди положительных элементов параллели главной диагонали, расположенной выше над диагональю/

```

package пэ2;
import static java.lang.Math.*;
public class пэ24 {
    public static void main(String[] args)
    {
        int n=4;
        int a[][]=new int [n][n];
        System.out.print("Матрица:"+"\n");
        for (int i=0;i<n;i++)
        {
            for (int j=0;j<n;j++)
            {
                a[i][j]=(int)Math.round(Math.random()*10-5);
                System.out.print( a[i][j]+ " ");
            }
            System.out.println();
        }
        int min=a[0][1];
        System.out.print("Минимальный элемент, лежащий выше главной диагонали:");
        for (int i=0;i<n;i++)
        {
            for (int j=0;j<n;j++)
            {
                if(i<j && a[i][j]<min && a[i][j]>-1)
                {
                    min=a[i][j];
                    System.out.print(min);
                }
            }
        }
    }
}

```

пэ2.пэ24 > main > for (int i = 0; i < n; i++) >

д ×

пэ2 (debug) × Консоль отладки ×

Матрица:

3 4 -1 0

0 0 1 -1

-1 5 -5 3

5 -2 1 -2

Минимальный элемент, лежащий выше главной диагонали:ОСБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

311.

3) Все элементы массива, меньшие заданного значения, и их номера записать в новые массивы. Вывести новые массивы на экран. Заданное значение определяется константой;

```

package javaapplication6;
import java.util.Scanner;

public class Main {
    static final int N = 2;
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter the length of the array: ");
        int arrlenght = in.nextInt();
        int[] Array = new int[arrlenght];
        int ElCounter = 0;
        for (int i = 0; i < Array.length; i++)
        {
            Array[i] = ((int) (Math.random() * 20) - 10);
        }
        for ( int i = 0; i < Array.length; ++i )
            if ( Array[i] < N )
                ElCounter++;
        int[] ArrNum = new int[ElCounter];
        int[] ArrEl = new int [ElCounter];
        int count = 0;
        int x = 0;
        for (int z = 0; z < ArrEl.length; z++)
        {
            for (x = 0; x < arrlenght; x++)
            {
                if (Array[x] < N) {
                    ArrEl[z] = Array[x];
                    ArrNum[z] = x;
                }
            }
            arrlenght = x;
            for (x = 0; x < arrlenght; x++)
                if (Array[x] < N)
                {
                    count = x;
                }
            arrlenght = count;
        }
        System.out.print("Source array: ");
        for ( int i = 0; i < Array.length; i++ )
            System.out.print(Array[i] + " ");
        System.out.print("\n" + "New array: ");
        for ( int i = 0; i < ElCounter; i++ )
    }
}

```

	<pre> 6 System.out.print (ArrEl[i] + " "); 7 System.out.print ("\n" + "Array of numbers: "); 8 for (int i = 0; i < ArrNum.length; i++) 9 System.out.print (ArrNum[i] + " "); 10 } 11 } </pre> <p>Вывод - JavaApplication6 (run) X</p> <pre> run: Enter the length of the array: 2 Source array: -9 4 New array: -9 Array of numbers: 0 СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 1 секун </pre>
312.	<p>Определить дополнительный массив разрешенных значений. Составить массив из элементов исходного, имеющих неразрешенные значения. Вывести результирующий массив на экран.</p> <p>Решение:</p> <pre> /* * To change this license header, choose License Headers in Project Properties. * To change this template file, choose Tools Templates * and open the template in the editor. */ package javatest2; /** * * @author User */ public class Javatest2 { public static int SIZE1 = 30; public static int MIN1 = 0; public static int MAX1 = 300; public static int MIN2 = 100; public static int MAX2 = 200; /** * @param args the command line arguments */ public static void main(String[] args) { // TODO code application logic here int[] values = new int[Javatest2.SIZE1]; int[] mass = new int[30]; System.out.print("Массив:\n"); for (int i = 0; i < Javatest2.SIZE1; i++) { values[i]=Javatest2.MIN1+(int)Math.round(Math.random()* (Javatest2.MAX1 - Javatest2.MIN1)); if(values[i]>MAX2 values[i]<MIN2) mass[i]=values[i]; System.out.print(mass[i]+"\\t"); } } } </pre>
313.	<p>Дан массив b (n). Переписать в массив C(n) положительные элементы массива b(n) деленные на 5. Затем упорядочить методом «выбора и перестановки» по возрастанию новый массив.</p> <p>Решение:</p> <pre> /* </pre>

	<pre> * To change this license header, choose License Headers in Project Properties. * To change this template file, choose Tools Templates * and open the template in the editor. */ package javatest2; /** * * @author User */ public class Javatest2 { /** * @param args the command line arguments */ public static void main(String[] args) { // TODO code application logic here double b[]={-1.0,2.0,-3.0,4.0,-5.0,6.0,-7.0,8.0,-9.0}; double c[]=new double[10]; for(int i=0; i<10;i++) { if(b[i]>0) c[i]=b[i]/5; System.out.print(c[i]+"\\t"); } } } </pre>
314.	<p>Определить матрицу и заполнить её случайными значениями. Построить вектор В, который возвращает – номер максимального значения в i-й строке. Решение:</p> <pre> /* * To change this license header, choose License Headers in Project Properties. * To change this template file, choose Tools Templates * and open the template in the editor. */ package javaapplication3; /** * * @author User */ public class JavaApplication3 { /** * @param args the command line arguments */ public static void main(String[] args) { // TODO code application logic here int[][] array = { {10,20,30}, {40,50,60}, {70,80,90} }; for (int i = 0; i < 3; i++) { for(int j = 0; j < 3; j++) { </pre>

	<pre> System.out.print(array[i][j] + " "); } System.out.println(); } System.out.println(); int index = 0; int k = 2; int max = array[0][0]; for (int i = k; ;) { for (int j = 0; j < 3; j++) { if(max<array[k][j]) { max = array[k][j]; index = j+1; } } break; } System.out.println(index); } } </pre>
315.	<p>Дан двумерный массив A, размером (nхn) (или квадратная матрица A). Найти количество положительных элементов её главной диагонали. Затем умножить побочную диагональ на найденное количество.</p> <p>Решение:</p> <pre> /* * To change this license header, choose License Headers in Project Properties. * To change this template file, choose Tools Templates * and open the template in the editor. */ package javaapplication3; /** * * @author User */ public class JavaApplication3 { /** * @param args the command line arguments */ public static void main(String[] args) { // TODO code application logic here int[][] A = { {-10,20,30}, {40,50,60}, {70,80,-90} }; for (int i = 0; i < 3; i++) { for(int j = 0; j < 3; j++) { System.out.print(A[i][j] + " "); } System.out.println(); } } } </pre>

```

}
System.out.println();
int k = 0;
for (int i = 0; i < 3; i++)
{
    if (A[i][i] > 0)
    {
        k++;
    }
}
System.out.println("Количество положительных элементов главной диагонали матрицы: " + k);
int q = 0;
int answ = 1;
for (int i = 0; i < 3; i++)
{
    q = A[i][3-i-1]*k;
    answ = answ*q;
}
System.out.println("Ответ на задание: " + answ);
}
}

```

316. Получить практические навыки разработки программ использованием объектно-ориентированного подхода на языке Java, создавать классы и объекты.

Вариант 4). Гостиница

Room – комната:

Свойства:

- Id – идентификатор;
 - codeNumbers – Код номера;
 - numberPeople – Количество человек;
 - comfortType – Комфортность;
 - price – цена.
- } Конструктор

```

package pkg1;
import static java.lang.Math.*;
public class Main {
    public static void main(String[] args) {
        Room room = new Room(1234, 56, 15);
        room.setprice(10000);
        room.setcomfortType(8);
        System.out.println(room.toString());
    }
}

class Room
{
    private int Id;
    private int codeNumbers;
    private int numberPeople;
    private int comfortType;
    private int price ;
    public Room (int Id, int codeNumbers, int numberPeople) {
        this.Id = Id;
        this.codeNumbers = codeNumbers;
        this.numberPeople = numberPeople;
    }

    public int getId() {
return Id;
}
}
public void setId (int Id) {

```

```

this.Id = Id;
}
public int getcodeNumbers() {
return codeNumbers;
}
public void setcodeNumbers (int codeNumbers) {
this.codeNumbers = codeNumbers;
}
public int getnumberPeople() {
return numberPeople;
}
public void setnumberPeople (int numberPeople) {
this.numberPeople = numberPeople;
}
public int getprice() {
return price;
}
public void setprice (int price) {
this.price = price;
}
public int getcomfortType() {
return comfortType;
}
public void setcomfortType (int comfortType) {
this.comfortType = comfortType;
}
public String toString()
{
return "Id: " + Id + " \n Код номера: " + codeNumbers + " \n Количество человек: "+ numberPeople + " \n
Комфортность:"+comfortType+" \n Цена: "+price;
}
}

```

317.

Вариант 8). Супермаркет

Supermarket:

Свойства:

- nameotdela – название отдела;
- productCode – Код товара;
- name – Наименование товара;
- cuntry – страна-производитель;
- retailPrice – Розничная цена;
- namesource – Поставщик;

} Конструктор

Листинг программы
 Файл Supermarket.java
 package practic_work_3;

```

public class Supermarket {
private int nameotdela;
private int productCode;
private String name;
private String cuntry;
private int retailPrice;
private String namesource;
public Supermarket(
int nameotdela,
int productCode,
String name,
String cuntry,
int retailPrice,
String namesource

```

```

        ) {
            this.nameotdela = nameotdela;
            this.productCode = productCode;
            this.name = name;
            this.cuntry = cuntry;
            this.retailPrice = retailPrice;
            this.namesource = namesource;
        }
    public int getnameotdela() {
        return nameotdela;
    }

    public void setnameotdela(int nameotdela) {
        this.nameotdela = nameotdela;
    }

    public int getproductCode() {
        return productCode;
    }
    public void setproductCode(int productCode) {
        this.productCode = productCode;
    }
    public String getname() {
        return name;
    }
    public void setname(String name) {
        this.name = name;
    }
    public String getcuntry() {
        return cuntry;
    }
    public void setcuntry(String cuntry) {
        this.cuntry = cuntry;
    }
    public int getretailPrice() {
        return retailPrice;
    }
    public void setretailPrice(int retailPrice) {
        this.retailPrice = retailPrice;
    }
    public String getnamesource() {
        return namesource;
    }
    public void setnamesource(String namesource) {
        this.namesource = namesource;
    }

    public String tooString() {
        return namesource + " привёз товар " + name + ", с кодом товара " + productCode + ", страны
производителя " + cuntry + ", \nс розничной ценой "
+ retailPrice + ", в отдел с номером " + nameotdela + ".";
    }
}

Файл TestSupermarket.java
package practic_work_3;

import practic_work_3.*;

```

	<pre>public class TestSupermarket { public static void main(String[] args) { Supermarket supermarket = new Supermarket(1,1234,"Картофель","Беларусь",500,"Иванов"); System.out.println(supermarket.toString()); System.out.println("\n" + supermarket); } }</pre>
318.	<p>Вариант 6). Успеваемость студентов ВУЗА Students – Студент: Id_studenta – номер зачетной книжки Fam – фамилия Name – имя Groupa – группа Department – кафедра discipline- Дисциплина mark- Оценка NameTeacher- Фамилия преподавателя</p> <p style="text-align: right;">} Конструктор</p> <pre>public class Student { private String Students; private int Id_studenta; private String Fam; private String Name; private int Groupa; private String Department; private String discipline; private int mark; private String NameTeacher; public Student(String Students, int Id_studenta, String Fam, String Name, int Groupa, String Department, String discipline, int mark, String NameTeacher) { this.Students = Students; this.Id_studenta = Id_studenta; this.Fam = Fam; this.Name = Name; this.Groupa = Groupa; this.Department = Department; this.discipline = discipline; this.mark = mark; this.NameTeacher = NameTeacher; } public String getStudents() { return Students; } }</pre>

```

public void SetStudent(String Students) {
    this.Students = Students;
}

public int getIdStudent() {
    return Id_studenta;
}

public void SetIdStudent(int Id_studenta) {
    this.Id_studenta = Id_studenta;
}

public String getFam() {
    return Fam;
}

public void SetFam(String Fam) {
    this.Fam = Fam;
}

public String getName() {
    return Name;
}

public void SetName(String Name) {
    this.Name = Name;
}

}

public int getGroupa() {
    return Groupa;
}

public void SetGroupa(int Groupa) {
    this.Groupa = Groupa;
}

public String getDepartment() {
    return Department;
}

public void SetDepartment(String Department) {
    this.Department = Department;
}

public String getDisipline() {
    return discipline;
}

public void SetDisipline(String discipline) {
    this.discipline = discipline;
}

public int getMark() {
    return mark;
}

public void SetMark(int mark) {
    this.mark = mark;
}

public String getNameTeacher() {
    return NameTeacher;
}

public void SetNameTeacher(String NameTeacher) {
    this.NameTeacher = NameTeacher;
}

public String tooString() {
    return "\n" + Students + "\nномер зачетной книжки: " + Id_studenta +
        "\nфамилия: " + Fam + "\nИмя: " + Name + "\nГруппа: " + Groupa + "\nКафедра: " +
        Department + "\nДисциплина: " + discipline + "\nОценка: " + mark +
        "\nфамилия преподавателя: " + NameTeacher;
}

public static void main(String[] args) {
    Student student = new Student(
        "Студент №321", 205696, "Лесных", "Захар", 203,
        "Информационных технологий", "Прикладная информатика",
        5, "Титова" );

    System.out.println(student.tooString());
    System.out.println("\n" + student);
}
}

```

319. Создать программу на языке Java для определения класса в некоторой предметной области. Описать свойства, конструктор, методы геттеры/сеттеры, перекрыть метод toString() для вывода полной информации об объекте в отформатированном виде:
 Вариант 11). Продажа путевок Tourist: Свойства: – код путевки; – фамилия клиента; – название пансионата; – номер; – вид жилья; – дата заезда; – дата выезда; – количество человек; – цена;

```

Код программы
package задание.pkg3;

/**
 *
 * @author ПК
 */
public class Tourist {
    private int code;
    private String name;
    private String title;
    private int Number;
    private String typeofhousing;
    private String arrivaldate;
    private String departuredate;
    private int numberofpeople;
    private int price;

    public Tourist(int code, String name, String title, int Number,String typeofhousing,String arrivaldate,String
    departuredate,int numberofpeople,int price)
    {
        this.code = code;
        this.name = name;
        this.title = title;
        this.Number = Number;
        this.typeofhousing = typeofhousing;
        this.arrivaldate = arrivaldate;
        this.departuredate = departuredate;
        this.numberofpeople = numberofpeople;
        this.price = price;
    }
    public int getCode() {
        return code;
    }
    public void setCode(String name) {
        this.code = code;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String name) {
        this.title = title;
    }
    public int getNumber() {
        return Number;
    }
    public void setNumber(int age) {
        this.Number = Number;
    }
    public String gettypeofhousing() {
        return typeofhousing;
    }
    public void settypeofhousing(String name) {
        this.typeofhousing = typeofhousing;
    }
    public String getarrivaldate() {
        return arrivaldate;
    }

```

```

}
public void setarrivaldate(String name) {
this.arrivaldate = arrivaldate;
}
public String getdeparturedate() {
return departuredate;
}
public void setdeparturedate(String name) {
this.departuredate = departuredate;
}
public int numberofpeople() {
return numberofpeople;
}
public void numberofpeople(String name) {
this.numberofpeople = numberofpeople;
}
public int price() {
return price;
}
public void price(String name) {
this.price = price;
}
public String toString()
{
return " Код путёвки " + code + " Фамилия клиента " + name + " наименование пансионата "+title+"
номер "+Number+" Вид жилья "+typeofhousing+" дата заезда "+arrivaldate+" дата выезда
"+departuredate+" количество человек "+numberofpeople+" цена "+price+" рублей.";
}
public static void main(String[] args) {
Tourist tourist = new Tourist(3415,"Патрушов","Лазурный час", 45,
"Квартира","13,05,2014","13,07,2014",2,8455);
System.out.println(tourist.toString());
System.out.println(tourist);
}
}

```

320.

Вариант 9). Военный состав

Command:
Свойства:

- фамилия;
- рота;
- звание;
- дата рождения;
- дата поступления на службу;
- часть;

} Конструктор

```

package prz3;

public class PRZ3 {

    public class command{
        public String surname;
        public String rota;
        public String rank;
        public String dateOfBirth;
        public String dateOfReceipt;
        public String militaryUnit;

        public command(String surname, String rota, String rank, String dateOfBirth, String dateOfReceipt,
String militaryUnit) {
            this.surname = surname;
            this.rota = rota;

```

```

        this.rank = rank;
        this.dateOfBirth = dateOfBirth;
        this.dateOfReceipt = dateOfReceipt;
        this.militaryUnit = militaryUnit;
    }

    public String getSurname() {
        return surname;
    }

    public String getRota() {
        return rota;
    }

    public String getRank() {
        return rank;
    }

    public String getDateOfBirth() {
        return dateOfBirth;
    }

    public String getDateOfReceipt() {
        return dateOfReceipt;
    }

    public String getMilitaryUnit() {
        return militaryUnit;
    }

    public void setSurname(String surname) {
        this.surname = surname;
    }

    public void setRota(String rota) {
        this.rota = rota;
    }

    public void setRank(String rank) {
        this.rank = rank;
    }

    public void setDateOfBirth(String dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
    }

    public void setDateOfReceipt(String dateOfReceipt) {
        this.dateOfReceipt = dateOfReceipt;
    }

    public void setMilitaryUnit(String militaryUnit) {
        this.militaryUnit = militaryUnit;
    }

    @Override
    public String toString() {
        return "command{" + "surname=" + surname + ", rota=" + rota + ", rank=" + rank + ", dateOfBirth="
+ dateOfBirth + ", dateOfReceipt=" + dateOfReceipt + ", militaryUnit=" + militaryUnit + '}';
    }
}
}

```

321.

Вариант 11). Продажа путевок

Tourist:

Свойства:

- код путевки;
- фамилия клиента;
- название пансионата;
- номер;
- вид жилья;
- дата заезда;
- дата выезда;
- количество человек;
- цена;

} Конструктор

```

public class Tourist {
    private int code;
    private String surname;
    private String NameOfPansionat;
    private int number;
    private String TypeOfHousing;
    private String zaezd;
    private String viezd;
    private int count;
    private int price;

    public Tourist(int code, String surname,String NameOfPansionat, int num-
ber, String TypeOfHousing, String zaezd, String viezd, int count, int price) {
        this.code = code;
        this.surname = surname;
        this.NameOfPansionat = NameOfPansionat;
        this.number = number;
        this.TypeOfHousing = TypeOfHousing;
        this.zaezd = zaezd;
        this.viezd = viezd;
        this.count = count;
        this.price = price;
    }
    public int getCode() {
        return code;
    }
    public void setCode(int code) {
        this.code = code;
    }
    public String getSurname() {
        return surname;
    }
    public void setSurname(String surname) {
        this.surname = surname;
    }
    public String getNameOfPansionat() {
        return NameOfPansionat;
    }
    public void setNameOfPansionat(String NameOfPansionat) {
        this.NameOfPansionat = NameOfPansionat;
    }
    public int getNumber() {
        return number;
    }
    public void setNumber(int number) {
        this.number = number;
    }
    public String getTypeOfHousing() {
        return TypeOfHousing;
    }

```

```

    }
    public void setTypeOfHousing(String TypeOfHousing) {
        this.TypeOfHousing = TypeOfHousing;
    }
    public String getZaezd() {
        return zaezd;
    }
    public void setZaezd(String zaezd) {
        this.zaezd = zaezd;
    }

    public String getViezd() {
        return viezd;
    }
    public void setViezd(String viezd) {
        this.viezd = viezd;
    }

    public int getCount() {
        return count;
    }
    public void setCount(int count) {
        this.count = count;
    }

    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }

    public String toString()
    {
        return "Код: " + code + "    " + "Фами-
        лия: " + surname + "    " + "Пансионат: " + NameOfPansionat + "    " + "Номер: "
        + number + "    " + "Жилье: " + TypeOfHousing + "    " + "Дата заезда: " + zaezd +
        "    " + "Дата выезда: " + viezd + "    " + "Количество человек: " + count + "    "
        + "Цена: " + price; }}

    public class TestTourist {
    public static void main(String[] args) {
        Tourist tourist = new Tourist(11, "Белов", "Белград", 17, "Комната",
        "17.01.2021", "25.02.2022", 2, 40000);
        System.out.println(tourist.toString());}}

```

322.

Вариант 1). Записная книжка контактов.

Contact – запись информации о контакте в записную книжку.

Свойства:

- Id – идентификатор контакта;
- first-Name – имя;
- lastName – фамилия;
- address – адрес;

} Конструктор

59

-
- phone – телефон;
 - note – запись о контакте.

/*

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

package javaapplication3;

/**

*

* @author windows

*/

public class JavaApplication3 {

```

public class Contact {

```

```

    private String firstName;

```

```

    private String address;

```

```

    private String note;

```

```

    private int phone;

```

```

    public Contact(String firstName, String address, String note, int phone) {

```

```

        this.firstName = firstName;

```

```

        this.address = address;

```

```

        this.note = note;

```

```

        this.phone = phone;

```

```

    }

```

```

    public String getFirstName() {

```

```

        return firstName;

```

```

    }

```

```

    public String getAddress() {

```

```

        return address;

```

```

    }

```

```

    public String getNote() {

```

```

        return note;

```

```

    }

```

```

    public int getPhone() {

```

```

        return phone;

```

```

    }

```

```

    public void setFirstName(String firstName) {

```

```

        this.firstName = firstName;

```

```

    }

    public void setAddress(String address) {
        this.address = address;
    }

    public void setNote(String note) {
        this.note = note;
    }

    public void setPhone(int phone) {
        this.phone = phone;
    }

    @Override
    public String toString() {
        return "Contact{" + "firstName=" + firstName + ", address=" + address + ", note=" + note + ",
phone=" + phone + '}';
    }
}
}

```

323.

Создать программу на языке Java для определения класса в некоторой предметной области. Описать свойства, конструктор, методы геттеры/сеттеры, переопределить метод toString() для вывода полной информации об объекте в отформатированном виде:

Вариант 1). Записная книжка контактов.

Contact – запись информации о контакте в записную книжку.

Свойства:

- Id – идентификатор контакта;
 - first-Name – имя;
 - lastName – фамилия;
 - address – адрес;
- } Конструктор

59

-
- phone – телефон;
 - note – запись о контакте.

```
package практика.pkg3;
```

```

public class Практика3 {
    public class Contact {
        private String Id;
        private String firstName;
        private String lastName;
        private String address;
        private int phone;
        private String note;

        public Contact(String Id, String firstName, String lastName, String address, int phone, String note) {
            this.Id = Id;
            this.firstName = firstName;
            this.lastName = lastName;
            this.address = address;
            this.phone = phone;
            this.note = note;
        }
    }
}

```

```

    }

    public String getId() {
        return Id;
    }

    public void setId(String Id) {
        this.Id = Id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public int getPhone() {
        return phone;
    }

    public void setPhone(int phone) {
        this.phone = phone;
    }

    public String getNote() {
        return note;
    }

    public void setNote(String note) {
        this.note = note;
    }

    @Override
    public String toString() {
        return "Contact{" + "Id=" + Id + ", firstName=" + firstName + ", lastName=" + lastName + ",
address=" + address + ", phone=" + phone + ", note=" + note + '}';
    }

}
}

```

324.

Вариант 3). Телепрограмма.

Show - передача:

Свойства:

- author – ведущий;
- name – название;
- description – описание;
- periodType – периодичность (1 – ежедневно; 2 – еженедельно; 3 – ежемесе-
сячно).

} Конструктор

// Вариант 3

```
public class Show {
    private String author;
    private String name;
    private String description;

    private int periodType = 1;

    public Show(String author, String name, String description) {
        this.author = author;
        this.name = name;
        this.description = description;
    }

    @Override
    public String toString() {
        return "Show{" +
            "author=" + author + "\" +
            ", name=" + name + "\" +
            ", description=" + description + "\" +
            ", periodType=" + periodType +
            '"';
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }
}
```

```

public int getPeriodType() {
    return periodType;
}

public void setPeriodType(int periodType) {
    if(periodType < 1 || periodType > 3){
        return;
    }
    this.periodType = periodType;
}
}

```

325.

Вариант 6). Успеваемость студентов ВУЗА

Students – Студент:

Id_studenta – номер зачетной книжки

Fam – фамилия

Name – имя

Groupa – группа

Department – кафедра

discipline- Дисциплина

mark- Оценка

NameTeacher- Фамилия преподавателя

} Конструктор

```

package com.company;
public class Main {
    public class Usp
    {
        int Id_studenta;
        String Fam;
        String Name;
        int Groupa;
        String Department;
        String discipline;
        int mark;
        String NameTeacher;

        @Override
        public String toString() {
            return "Usp{" +
                "Id_studenta=" + Id_studenta +
                ", Fam=" + Fam + "\" +
                ", Name=" + Name + "\" +
                ", Groupa=" + Groupa +
                ", Department=" + Department + "\" +
                ", discipline=" + discipline + "\" +
                ", mark=" + mark +
                ", NameTeacher=" + NameTeacher + "\" +
                '}'";
        }
        public Usp(int id_studenta, String fam, String name, int groupa, String department, String discipline, int
mark, String nameTeacher) {
            Id_studenta = id_studenta;
            Fam = fam;
            Name = name;

```

```

        Groupa = groupa;
        Department = department;
        this.discipline = discipline;
        this.mark = mark;
        NameTeacher = nameTeacher;
    }
    public void setId_studenta(int id_studenta) {
        Id_studenta = id_studenta;
    }

    public void setFam(String fam) {
        Fam = fam;
    }

    public void setName(String name) {
        Name = name;
    }

    public void setGroupa(int groupa) {
        Groupa = groupa;
    }

    public void setDepartment(String department) {
        Department = department;
    }

    public void setDiscipline(String discipline) {
        this.discipline = discipline;
    }

    public void setMark(int mark) {
        this.mark = mark;
    }

    public void setNameTeacher(String nameTeacher) {
        NameTeacher = nameTeacher;
    }

    public int getId_studenta() {
        return Id_studenta;
    }

    public String getFam() {
        return Fam;
    }

    public String getName() {
        return Name;
    }

    public int getGroupa() {
        return Groupa;
    }

    public String getDepartment() {
        return Department;
    }

    public String getDiscipline() {

```

	<pre> return discipline; } public int getMark() { return mark; } public String getNameTeacher() { return NameTeacher; } } } </pre>
<p>326.</p>	<pre> Command: Свойства: - фамилия; - рота; - звание; - дата рождения; - дата поступления на службу; - часть; </pre> <p style="text-align: right;">} Конструктор</p> <pre> class Command { private String surname; private int company; private String rank; private Date birthday; private Date admission_day; private String part; public Command(String surname, int company, String rank, Date birthday) { this.surname = surname; this.company = company; this.rank = rank; this.birthday = birthday; } public String getSurname() { return this.surname; } public void setSurname(String surname) { this.surname = surname; } public int getCompany() { return this.company; } public void setCompany(int company) { this.company = company; } public String getRank() { return this.rank; } public void setRank(String rank) { this.rank = rank; } } </pre>

	<pre> } public Date getBirthday() { return this.birthday; } public void setBirthday(Date birthday) { this.birthday = birthday; } public Date getAdmissionDay() { return this.admission_day; } public void setAdmissionDay(Date admission_day) { this.admission_day = admission_day; } public String getPart() { return this.part; } public void setPart(String part) { this.part = part; } public String toString() { return "Данные о служащем:\n" + "Фамилия: " + surname + ";\n" + "Номер полка: " + company + ";\n" + "Звание: " + rank + ";\n" + "Дата рождения: " + birthday + ";\n" + "Дата поступления на службу: " + admission_day + ";\n" + "Название воинской части:" + part + ".\n"; } } public class Main { public static void main(String[] args) { Command command = new Command("Иванов", 5, "Младший лейтенант", new Date(1998, 12, 1)); command.setAdmissionDay(new Date(2013, 12, 3)); command.setPart("Воинская часть №5"); String surname = command.getSurname(); System.out.println(surname + '\n'); System.out.print(command.toString()); } } </pre>
327.	<p>Создать программу на языке Java для определения класса в некоторой предметной области. Описать свойства, конструктор, методы геттеры/сеттеры, перекрыть метод toString() для вывода полной информации об объекте в отформатированном виде:</p>

Вариант 7). Деканат

NameFaculty - факультет

Room – аудитория

corps - корпус

Telephone – контактный телефон

NameDean – фамилия декана

} Конструктор

Листинг программы

```
public class Deanery {  
  
    private String nameFaculty;  
  
    private int room;  
  
    private String corps;  
  
    private String telephone;  
  
    private String nameDean;  
  
    public Deanery(  
        String nameFaculty,  
        int room,  
        String corps,  
        String telephone,  
        String nameDean  
    ) {  
  
        this.nameFaculty = nameFaculty;  
  
        this.room = room;  
  
        this.corps = corps;  
  
        this.telephone = telephone;  
  
        this.nameDean = nameDean;  
  
    }  
  
    public String getnameFaculty() {  
  
        return nameFaculty;  
  
    }  
  
    public void setnameFaculty(String nameFaculty) {  
  
        this.nameFaculty = nameFaculty;  
  
    }  
  
    public int getroom() {
```

```

        return room;
    }

    public void setroom(int room) {
        this.room = room;
    }

    public String getcorps() {
        return corps;
    }

    public void setcorps(String corps) {
        this.corps = corps;
    }

    public String gettelephone() {
        return telephone;
    }

    public void settelephone(String telephone) {
        this.telephone = telephone;
    }

    public String getnameDean() {
        return nameDean;
    }

    public void setnameDean(String nameDean) {
        this.nameDean = nameDean;
    }

    public String tooString() {
        return "\n-----
-----" +
                "\n\t\tВоронежский государственный университет инженерных
технологий" +
                "\n-----
-----" +
                "\nФакультет: " + nameFaculty + "\nАудитория: " + room +
                "\nКорпус: " + corps + "\nКонтактный телефон: " +

```

```

        telephone + "\nФамилия декана: " + nameDean;
    }
}

public class TestDeanery {

    public static void main(String[] args) {

        Deanery deanery = new Deanery(

            "Управления и информатики в технологических системах",

            217, "Учебно-лабораторный", "+7(473)255-38-17",

            "Скрыпников");

        System.out.println(deanery.toString());

        System.out.println("\n" + deanery);

    }

}

```

328.

Вариант 11). Продажа путевок

Tourist:

Свойства:

- код путевки;
- фамилия клиента;
- название пансионата;
- номер;
- вид жилья;
- дата заезда;
- дата выезда;
- количество человек;
- цена;

} Конструктор

p:

p

```

private int voucherCode;
private String lastName;
private String nameBoardingHouse;
private int number;
private String typeHousing;
private String dateArrival;
private String dateDeparture;
private int countPeople;
private int price;

public Tourist(
    int voucherCode,
    String lastName,
    String nameBoardingHouse,
    int number,
    String typeHousing,
    String dateArrival,
    String dateDeparture,
    int countPeople,
    int price
) {
    this.voucherCode = voucherCode;
    this.lastName = lastName;
    this.nameBoardingHouse = nameBoardingHouse;
    this.number = number;
    this.typeHousing = typeHousing;
    this.dateArrival = dateArrival;

```

```

        this.dateDeparture = dateDeparture;
        this.countPeople = countPeople;
        this.price = price;
    }
    public int getvoucherCode() {
        return voucherCode;
    }
    public void setvoucherCode(int voucherCode) {
        this.voucherCode = voucherCode;
    }
    public String getlastName() {
        return lastName;
    }
    public void setlastName(String lastName) {
        this.lastName = lastName;
    }
    public String getnameBoardingHouse() {
        return nameBoardingHouse;
    }
    public void setnameBoardingHouse(String nameBoardingHouse) {
        this.nameBoardingHouse = nameBoardingHouse;
    }
    public int getnumber() {
        return number;
    }
    public void setnumber(int number) {
        this.number = number;
    }
    public String gettypeHousing() {
        return typeHousing;
    }
    public void settypeHousing(String typeHousing) {
        this.typeHousing = typeHousing;
    }
    public String getdateArrival() {
        return dateArrival;
    }
    public void setvoucherCode(String dateArrival) {
        this.dateArrival = dateArrival;
    }
    public String getdateDeparture() {
        return dateDeparture;
    }
    public void setdateDeparture(String dateDeparture) {
        this.dateDeparture = dateDeparture;
    }
    public int getcountPeople() {
        return countPeople;
    }
    public void setcountPeople(int countPeople) {
        this.countPeople = countPeople;
    }
    public int getprice() {
        return price;
    }
    public void setprice(int price) {
        this.price = price;
    }
    public String tooString() {
        return lastName + " с кодом путевки " + voucherCode + " с " + dateArrival + " по " +
dateDeparture + " \nпроживал в "
+ typeHousing + " номер " + number + " пансионата " + nameBoardingHouse + " вместе с " +
countPeople + " товарищами" + ", \nзаплатив за эту услугу " + price + "

```

	<pre> рублей."; } } Файл TestTourist.java package third_practic_work; import third_practic_work.*; public class TestTourist { public static void main(String[] args) { Tourist tourist = new Tourist(5432,"Тарасов", "Ласточка", 5, "комнате", "23.11.2017", "23.12.2017", 2, 13000); System.out.println(tourist.toString()); System.out.println("\n" + tourist); } } </pre>
--	---

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

4.1. Рейтинговая система оценки осуществляется в течение всего семестра при проведении аудиторных занятий, показателем ФОС является текущий опрос в виде собеседования, за каждый правильный ответ обучающийся получает 5 баллов (зачтено - 5, не зачтено - 0). Максимальное число баллов по результатам текущей работы в семестре 50.

4.2. Бальная система служит для получения экзамена по дисциплине.

Максимальное число баллов за семестр – 100.

Максимальное число баллов по результатам текущей работы в семестре – 50.

Максимальное число баллов на зачете – 50.

Минимальное число баллов за текущую работу в семестре – 30.

Обучающийся, набравший в семестре менее 30 баллов может заработать дополнительные баллы, отработав соответствующие разделы дисциплины или выполнив обязательные задания, для того чтобы быть допущенным до экзамена.

Обучающийся, набравший за текущую работу менее 30 баллов, т.к. не выполнил всю работу в семестре по объективным причинам (болезнь, официальное освобождение и т.п.) допускается до экзамена, однако ему дополнительно задаются вопросы на собеседовании по разделам, выносимым на экзамен.

В случае неудовлетворительной сдачи экзамена обучающемуся предоставляется право повторной сдачи в срок, установленный для ликвидации академической задолженности по итогам соответствующей сессии. При повторной сдаче экзамена количество набранных баллов на предыдущем экзамене не учитывается.

Экзамен может проводиться в виде тестового задания и кейс-задания или собеседования и кейс-заданий и/или задач.

Для получения оценки «отлично» суммарная бально-рейтинговая оценка по результатам работы в семестре и на экзамене должна составлять 90 и выше баллов;

- оценки «хорошо» суммарная бально-рейтинговая оценка студента по результатам работы в семестре и на экзамене должна составлять от 75 до 89,99 баллов;

- оценки «удовлетворительно» суммарная бально-рейтинговая оценка обучающегося по результатам работы в семестре и на экзамене должна составлять от 60 до 74,99 баллов;

- оценки «неудовлетворительно» суммарная бально-рейтинговая оценка обучающегося по результатам работы в семестре и на экзамене должна составлять менее 60 баллов.

5 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания для каждого результата обучения по дисциплине/практике

Результаты обучения по этапам формирования компетенций	Предмет оценки (продукт или процесс)	Показатель оценивания	Критерии оценивания сформированности компетенций	Шкала оценивания	
				Академическая оценка или баллы	Уровень освоения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач					
УМЕТЬ: умеет критически оценивать достоинства и недостатки, а также сильные и слабые стороны своей профессиональной деятельности	Собеседование (экзамен)		Обучающийся полностью раскрыл содержание материала в объеме, предусмотренном программой, изложил материал грамотным языком в определенной логической последовательности	отлично	освоена (повышенный)
			Обучающийся твердо знает материал, грамотно и по существу излагает его, но допускает в ответе некоторые неточности	хорошо	освоена (повышенный)
			Обучающийся неполно или непоследовательно раскрыл содержание материала, но показал общее понимание вопроса, недостаточно правильные формулировки базовых понятий	удовлетворительно	освоена (базовый)
			Обучающийся не раскрыл содержание материала, допускает грубые ошибки в формулировках основных понятий дисциплины	не удовлетворительно	не освоена (недостаточный)
ЗНАТЬ: основные алгоритмы типовых численных методов решения математических задач	Тестовое задание	Результат тестирования	Набрано менее 50% правильных ответов	не удовлетворительно	не освоена (недостаточный)
			Набрано 50% - 74,99% правильных ответов	удовлетворительно	освоена (базовый)
			Набрано 75% - 89,99% правильных ответов	хорошо	освоена (повышенный)
			Набрано 90% - 100% правильных ответов	отлично	освоена (повышенный)
	Собеседование (экзамен)	Знание типовых численных методов решения математических задач	Обучающийся полностью раскрыл содержание материала в объеме, предусмотренном программой, изложил материал грамотным языком в определенной логической последовательности	отлично	освоена (повышенный)
			Обучающийся твердо знает материал, грамотно и по существу излагает его, но допускает в ответе некоторые неточности	хорошо	освоена (повышенный)
			Обучающийся неполно или непоследовательно раскрыл содержание материала, но показал общее понимание вопроса, недостаточно правильные формулировки базовых понятий	удовлетворительно	освоена (базовый)
			Обучающийся не раскрыл содержание материала, допускает грубые ошибки в формулировках основных понятий дисциплины	не удовлетворительно	не освоена (недостаточный)

УМЕТЬ: умеет применять вероятностно-статистический подход к оценке точности средств измерений, испытаний и качества продукции и технологических процессов	Защита практических работ	Применение вероятностно-статистического подхода к оценке точности средств измерений, испытаний и качества продукции и технологических процессов	Защита по практической работе соответствует теме	зачтено	освоена (повышенный)
			Защита по практической работе не соответствует теме	не зачтено	не освоено (недостаточный)
ВЛАДЕТЬ: эффективно используется математическим аппаратом, необходимым для осуществления профессиональной деятельности	Кейс-задания	Содержание решения кейс-задания	Обучающийся разобрался в предложенной конкретной ситуации, самостоятельно решил поставленную задачу на основе полученных знаний	зачтено	освоена (повышенный)
			Обучающийся не разобрался в сложившейся ситуации, не выявил причины случившегося и не предложил вариантов решения	не зачтено	освоена (повышенный)
ПК-14 способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач					
ЗНАТЬ: знает методы поиска и анализа необходимой информации, технических данных, показателей и результатов работы	Тестовое задание	Результат тестирования	Набрано менее 50% правильных ответов	не удовлетворительно	не освоена (недостаточный)
			Набрано 50% - 74,99% правильных ответов	удовлетворительно	освоена (базовый)
			Набрано 75% - 89,99% правильных ответов	хорошо	освоена (повышенный)
			Набрано 90% - 100% правильных ответов	отлично	освоена (повышенный)
	Собеседование (эк-замен)	Знание методов поиска и анализа необходимой информации, технических данных, показателей и результатов работы	Обучающийся полностью раскрыл содержание материала в объеме, предусмотренном программой, изложил материал грамотным языком в определенной логической последовательности	отлично	освоена (повышенный)
			Обучающийся твердо знает материал, грамотно и по существу излагает его, но допускает в ответе некоторые неточности	хорошо	освоена (повышенный)
			Обучающийся неполно или непоследовательно раскрыл содержание материала, но показал общее понимание вопроса, недостаточно правильные формулировки базовых понятий	удовлетворительно	освоена (базовый)
			Обучающийся не раскрыл содержание материала, допускает грубые ошибки в формулировках основных понятий дисциплины	не удовлетворительно	не освоена (недостаточный)
УМЕТЬ: умеет проводить анализ необходимой информации, технических данных,	Защита практических работ	проведение анализа необходимой информации, технических	Защита по практической работе соответствует теме	зачтено	освоена (повышенный)
			Защита по практической работе не со-	не зачтено	не освоено

показателей и результатов работы		данных, показателей и результатов работы	ответствует теме		(недостаточный)
ВЛАДЕТЬ: владеет навыками применения стандартных программных средств в области технического регулирования и метрологии, способен выполнять обобщение и систематизацию технических данных.	Кейс-задания	Содержание решения кейс-задания	Обучающийся разобрался в предложенной конкретной ситуации, самостоятельно решил поставленную задачу на основе полученных знаний	зачтено	освоена (повышенный)
			Обучающийся не разобрался в сложившейся ситуации, не выявил причины случившегося и не предложил вариантов решения	не зачтено	освоена (повышенный)
ПК-14 способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач					
ЗНАТЬ: знает методы поиска и анализа необходимой информации, технических данных, показателей и результатов работы	Тестовое задание	Результат тестирования	Набрано менее 50% правильных ответов	не удовлетворительно	не освоена (недостаточный)
			Набрано 50% - 74,99% правильных ответов	удовлетворительно	освоена (базовый)
			Набрано 75% - 89,99% правильных ответов	хорошо	освоена (повышенный)
			Набрано 90% - 100% правильных ответов	отлично	освоена (повышенный)
	Собеседование (эк-замен)	Знание методов поиска и анализа необходимой информации, технических данных, показателей и результатов работы	Обучающийся полностью раскрыл содержание материала в объеме, предусмотренном программой, изложил материал грамотным языком в определенной логической последовательности	отлично	освоена (повышенный)
			Обучающийся твердо знает материал, грамотно и по существу излагает его, но допускает в ответе некоторые неточности	хорошо	освоена (повышенный)
			Обучающийся неполно или непоследовательно раскрыл содержание материала, но показал общее понимание вопроса, недостаточно правильные формулировки базовых понятий	удовлетворительно	освоена (базовый)
			Обучающийся не раскрыл содержание материала, допускает грубые ошибки в формулировках основных понятий дисциплины	не удовлетворительно	не освоена (недостаточный)
УМЕТЬ: умеет проводить анализ необходимой информации, технических данных, показателей и результатов работы	Защита практических работ	проведение анализа необходимой информации, технических данных, показателей и результатов работы	Защита по практической работе соответствует теме	зачтено	освоена (повышенный)
			Защита по практической работе не соответствует теме	не зачтено	не освоено (недостаточный)
ВЛАДЕТЬ: владеет навыками применения стандартных программных средств в области технического регулирования и	Кейс-задания	Содержание решения кейс-задания	Обучающийся разобрался в предложенной конкретной ситуации, самостоятельно решил поставленную задачу на основе полученных знаний	зачтено	освоена (повышенный)

метрологии, способен выполнять обобщение и систематизацию технических данных.			Обучающийся не разобрался в сложившейся ситуации, не выявил причины случившегося и не предложил вариантов решения	не зачтено	освоена (повышенный)
---	--	--	---	------------	-------------------------