

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

09.03.02 Информационные системы и технологии

Направленность (профиль) подготовки

Моделирование и проектирование информационных технологий и систем

Квалификация выпускника

бакалавр

Воронеж

1. Цели и задачи дисциплины (модуля)

Целями освоения дисциплины является получение знаний о современном объектно-ориентированном языке программирования Java и овладение основными приемами программирования. Получение практических навыков работы по разработке программ на языке Java.

Дисциплина формирует компетенции у обучающихся в следующих областях профессиональной деятельности:

- 06 Связь, информационные и коммуникационные технологии (в сфере исследования, разработки, внедрения и сопровождения информационных технологий);

Дисциплина направлена на решение задач профессиональной деятельности следующих типов:

- производственно- технологический
- организационно- управленческий
- проектный.

Программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.02 «Информационные системы и технологии».

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

В результате освоения дисциплины в соответствии с предусмотренными компетенциями обучающийся должен:

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции
ПКв-3	способностью использовать технологии разработки объектов профессиональной деятельности в областях: управление технологическими процессами, химическая промышленность, пищевая промышленность, а также предприятия различного профиля и все виды деятельности в условиях экономики информационного общества	ИД-1 _{ПКв-3} . Знать: технологии разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.
		ИД-2 _{ПКв-3} . Уметь: применять технологии разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.
		ИД-3 _{ПКв-3} . Владеть: навыками использования технологий разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД-1 _{ПКв-3} . Знать: технологии разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.	Знает: синтаксис языка Java; основные типы данных языка Java;
	Умеет: анализировать код программы, написанный на языке Java; сопоставлять компоненты языка Java
	Владеет: средствами программирования на Java
ИД-2 _{ПКв-3} . Уметь: применять технологии разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.	Знает: принцип работы с каждым типом данных; управляющие конструкции языка
	Умеет: применять систему ввода/вывода на Java; осуществлять обработку массивов
	Владеет: приемами документирования кода программы; навыками создания консольных и оконных (GUI) приложений на Java
Владеть: навыками использования технологий разработки программ-	Знает: возможности языка и области применения Java – приложений; основные пакеты, классы, методы языка Java

ных продуктов, предназначенных для решения задач профессиональной деятельности.	Умеет: создавать консольные приложения на языке Java
	Владеет: приемами работы со встроенными типами данных; стандартными методами работы с файлами и каталогами; технологией поточного ввода/вывода

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

3.1. Дисциплина (модуль) _____ Программирование на Java
(наименование дисциплины (модуля))

относится к блоку 1 ОП и ее ФТД.В.02 Факультативы вариативная часть.

4. Объем дисциплины (модуля) и виды учебных занятий

Общая трудоемкость дисциплины (модуля) составляет 2 зачетных единиц.

Виды учебной работы	Всего часов	Семестр
		4
	акад.	акад.
Общая трудоемкость дисциплины (модуля)	72	72
Контактная работа, в т.ч. аудиторные занятия:	30,85	30,85
Лекции	15	15
Практические занятия (ПЗ)	15	15
Семинары (С)	-	-
Лабораторные работы (ЛБ)	-	-
Консультации текущие	0,75	0,75
Проведение консультаций перед экзаменом		
Виды аттестации (зачет, экзамен)	0,1	0,1
Самостоятельная работа:	41,15	41,15
Проработка материалов по конспекту лекций	10	10
Проработка материалов по учебникам, учебным пособиям	17,65	17,65
Курсовой проект (работа)		
Реферат		
и (или) другие виды самостоятельной работы	13,5	13,5
Подготовка к экзамену (контроль)	-	-

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, час
1	Основы Java	Основы Java	9,15
2	ООП в Java, наследование	ООП в Java, наследование	10
3	Операции в Java	Операции в Java	10
4	Массивы в Java	Массивы в Java	10
5	Структура Java программ	Структура Java программ	10
6	Коллекции элементов	Коллекции элементов	12
7	Исключительные ситуации	Исключительные ситуации	10

<i>Консультации текущие</i>	0,9
<i>зачет</i>	0,1

5.2 Разделы дисциплины (модуля) и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции, час	ПЗ (или С), час	ЛР, час	СРО, час
1	Основы Java	2	2		5,15
2	ООП в Java, наследование	2	2		6
3	Операции в Java	2	2		6
4	Массивы в Java	2	2		6
5	Структура Java программ	2	2		6
6	Коллекции элементов	3	3		6
7	Исключительные ситуации	2	2		6
<i>Консультации текущие</i>					0,9
<i>зачет</i>					0,1

5.2.1 Лекции

№ п/п	Наименование раздела дисциплины	Тематика лекционных занятий	Трудоемкость, 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.	2
2	ООП в Java, наследование	ООП в Java, наследование. Объектно-ориентированный язык программирования. 5 принципов объектно-ориентированного подхода. Реализация принципов объектно-ориентированного подхода в Java.	2
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).	2
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).	2
5	Структура Java программ	Структура Java программ. Знакомство с библиотеками и пакетами. Использование пакетов в программах. Создание своих собственных пакетов	2
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 ситуа-	Исключительные ситуации. Обработка исключительных ситуаций (Exceptions). Пове-	2

	ции	дение программы при возникновении исключения. Структура и использование блока перехвата исключений. . Классы исключительных ситуаций. Механизм контроля перехвата исключений. Практическая работа. Блок fmaNy. Методы класса Throwable.	
--	-----	---	--

5.2.2 Практические занятия (семинары)

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

5.2.3 Лабораторный практикум не предусмотрен.

5.2.4 Самостоятельная работа обучающихся (СРО)

№ п/п	Наименование раздела дисциплины	Вид СРО	Трудоемкость, час
1	2	3	4
1	Основы Java	Проработка материалов по курсу лекций	5,15
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
2	ООП в Java, наследование	Проработка материалов по курсу лекций	6
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
3	Операции в Java	Проработка материалов по курсу лекций	6
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
4	Массивы в Java	Проработка материалов по курсу лекций	6
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
5	Структура Java программ	Проработка материалов по курсу лекций	6
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	
6	Коллекции элементов	Проработка материалов по курсу лекций	6
		Проработка материалов по учебнику	
		Расчетно-практическая работа	
		Оформление отчетов	

7	Исключительные ситуации	си-	Проработка материалов по курсу лекций	6
			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://npoad.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 отдельным комплектом и **входят в состав рабочей программы дисциплины (модуля).**

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 акад.	Семестр 4
Общая трудоемкость дисциплины (модуля)	72	72
Контактная работа, в т.ч. аудиторные занятия:	13,8	13,8
Лекции	4	4
Практические занятия (ПЗ)	4	4
Семинары (С)	-	-
Лабораторные работы (ЛБ)	-	-
Консультации текущие	1,5	1,5
Проведение консультаций перед экзаменом	-	-
Виды аттестации (зачет, экзамен)	зачет	
Самостоятельная работа:	58,6	58,6
Проработка материалов по конспекту лекций	28,6	28,6
Проработка материалов по учебникам, учебным пособиям	30	30
Курсовой проект (работа)	-	-
Реферат	-	-
и (или) другие виды самостоятельной работы	-	-
Контроль	3,9	3,9

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

по дисциплине

Программирование на Java

Требования к результатам освоения дисциплины

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции
ПКв-3	способностью использовать технологии разработки объектов профессиональной деятельности в областях: управление технологическими процессами, химическая промышленность, пищевая промышленность, а также предприятия различного профиля и все виды деятельности в условиях экономики информационного общества	ИД-1 _{ПКв-3} Знать: технологии разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.
		ИД-2 _{ПКв-3} Уметь: применять технологии разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.
		ИД-3 _{ПКв-3} Владеть: навыками использования технологий разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД-1 _{ПКв-3} Знать: технологии разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.	Знает: синтаксис языка Java; основные типы данных языка Java;
	Умеет: анализировать код программы, написанный на языке Java; сопоставлять компоненты языка Java
	Владеет: средствами программирования на Java
ИД-2 _{ПКв-3} Уметь: применять технологии разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.	Знает: принцип работы с каждым типом данных; управляющие конструкции языка
	Умеет: применять систему ввода/вывода на Java; осуществлять обработку массивов
	Владеет: приемами документирования кода программы; навыками создания консольных и оконных (GUI) приложений на Java
Владеть: навыками использования технологий разработки программных продуктов, предназначенных для решения задач профессиональной деятельности.	Знает: возможности языка и области применения Java – приложений; основные пакеты, классы, методы языка Java
	Умеет: создавать консольные приложения на языке Java
	Владеет: приемами работы со встроенными типами данных; стандартными методами работы с файлами и каталогами; технологией поточного ввода/вывода

1. 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	ПКв-3	Тестовые задания	1-185	Рубежный контроль
			Кейс-задания	186-211	Рубежный контроль
			Вопросы к зачету	212-259	Итоговый контроль
2	Структура Java программ Коллекции элементов Основы Java	ПКв-3	Тестовые задания	1-185	Рубежный контроль
			Кейс-задания	186-211	Рубежный контроль
			Вопросы к зачету	212-259	Итоговый контроль
3	ООП в Java, наследование Операции в Java Массивы в Java Структура Java программ	ПКв-3	Тестовые задания	1-185	Рубежный контроль
			Кейс-задания	186-211	Рубежный контроль
			Вопросы к зачету	212-259	Итоговый контроль

4	Коллекции элементов Основы Java ООП в Java, наследование	ПКв-3	Тестовое задание	1-185	Рубежный контроль
			Кейс-задания	186-211	Рубежный контроль
			Вопросы к зачету	212-259	Итоговый контроль
5	Операции в Java Массивы в Java Структура Java программ Коллекции элементов	ПКв-3	Тестовое задание	1-185	Рубежный контроль
			Кейс-задания	186-211	Рубежный контроль
			Вопросы к зачету	212-259	Итоговый контроль
6	Исключительные ситуации Основы Java ООП в Java, наследование Операции в Java	ПКв-3	Тестовое задание	1-185	Рубежный контроль
			Кейс-задания	186-211	Рубежный контроль
			Вопросы к зачету	212-259	Итоговый контроль
7	Массивы в Java, Создание клиентского приложения к СУБД	ПКв-3	Тестовое задание	1-185	Рубежный контроль
			Кейс-задания	186-211	Рубежный контроль
			Вопросы к зачету	212-259	Итоговый контроль

3. Оценочные средства для промежуточной аттестации

3.1 Тесты (тестовые задания)

№ задания	Тестовое задание
1.	Дан код: <pre>class Quest3 {public static void main(String s[]) { String args; System.out.print(args + s);}}</pre> Результатом компиляции кода будет? >> ошибка компиляции: переменная args используется до инициализации.
2.	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.
3.	Программы Java категория Приложение (application)? >> аналог "обычной" прикладной программы.
4.	Программы Java категория Апплет (applet)? >> специализированная программа с ограниченными возможностями, работающая в окне WWW-документа под управлением браузера.
5.	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.
6.	Программы Java категория Серверное приложение (Enterprise application)? >> предназначено для многократного использования на стороне сервера.
7.	Программы Java категория Библиотека (Java Class Library)? >> библиотека классов, либо NetBeansModule – модуль платформы NetBeans) – предназначена для многократного использования программами Java.

8.	Какие из следующих строк скомпилируются без ошибки? >> int i = 32565; - float f = 7.0; - char c = "z"; - byte b = 255; - boolean n = null;
9.	Какие из следующих строк скомпилируются без ошибки? >> int j = 'Ъ'; - float f = 7.0; - char c = "z"; - byte b = 255; - boolean n = null;
10.	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-");
11.	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-");
12.	Какие из следующих идентификаторов являются корректными? >> _int; - 2int; - int_#; - #int - _#int
13.	Назначение утилиты javac? >> Компилятор в режиме командной строки для программ, написанных на языке Java.
14.	Назначение утилиты java? >> Утилита для запуска в режиме командной строки откомпилированных программ-приложений.
15.	Назначение утилиты appletviewer? >> Утилита для запуска на исполнение и отладку апплетов без браузера. При этом не гарантируется работоспособность отлаженного апплета в браузере.
16.	Назначение утилиты jdb? >> Отладчик программ, написанных на языке Java.
17.	Назначение утилиты javadoc? >> Генератор документации по классом на основе комментариев, начинающихся с /**.
18.	Назначение утилиты jar? >> Создание и управление Java-архивами jar.
19.	Назначение утилиты javah? >> Генератор заголовочных файлов C/C++ для подключения к программам Java внешних библиотек C/C++ на основе интерфейса JNI.
20.	Назначение утилиты javap? >> Дизассемблер классов.
21.	Назначение утилиты extcheck? >> Утилита для обнаружения конфликтов между файлами архивов jar.
22.	Назначение утилиты native2ascii? >> Утилита для конвертации в режиме командной строки параметра, передаваемого в виде текста на национальном алфавите, в последовательность символов UNICODE.

23.	Управляющая последовательность: \a ? >> звонок.		
24.	Управляющая последовательность: \b ? >> возврат на один символ назад.		
25.	Управляющая последовательность: \f ? >> перевод на новую страницу.		
26.	Управляющая последовательность: \n ? >> перевод на новую строку.		
27.	Управляющая последовательность: \r ? >> возврат к началу строки.		
28.	Управляющая последовательность: \t ? >> горизонтальная табуляция.		
29.	Управляющая последовательность: \v ? >> вертикальная табуляция.		
30.	Управляющая последовательность: \' ? >> кавычка.		
31.	Управляющая последовательность: \" ? >> двойные кавычки.		
32.	Управляющая последовательность: \\ ? >> обратная косая черта.		
33.	Управляющая последовательность: \u ? >> начало кодировки символа Unicode.		
34.	Оператор ~ ? >> Оператор побитового дополнения (побитовое “не”).		
35.	Оператор ^ ? >> Оператор “исключающее или” (XOR).		
36.	Оператор & ? >> Оператор “побитовое и” (AND).		
37.	Оператор ? >> Оператор “побитовое или” (OR).		
38.	Оператор ! ? >> Оператор “НЕ”.		
39.	Оператор && ? >> Оператор “логическое И” (AND).		
40.	Оператор ? >> Оператор “логическое ИЛИ” (OR).		
41.	Оператор << ? >> Оператор левого побитового сдвига.		
42.	Оператор >>> ? >> Оператор беззнакового правого побитового сдвига.		
43.	Оператор >> ? >> Оператор правого побитового сдвига с сохранением знака отрицательного числа.		
44.	Для чего предназначен пакет java.applet? >> Классы, необходимые для создания апплетов.		
45.	Для чего предназначен пакет java.awt? >> Графический пользовательский интерфейс на базе AbstractWindowsToolkit.		
46.	Для чего предназначен пакет java.awt.color? >> Пространства цветов для AWT.		
47.	Для чего предназначен пакет java.awt.datatransfer? >> Передача данных внутри программы и между программами.		
48.	Для чего предназначен пакет java.awt.dnd? >> Поддержка технологии drag-and-drop.		
49.	Для чего предназначен пакет java.awt.event? >> Обработка событий AWT.		
50.	Для чего предназначен пакет java.awt.im? >> Поддержка нестандартных методов ввода текста.		

51.	Для чего предназначен пакет java.awt.im.spi? >> Интерфейсы для создания нестандартных методов ввода текста.
52.	Для чего предназначен пакет java.lang? >> Базовые классы языка Java.
53.	Для чего предназначен пакет java.lang.ref? >> Управление ссылками на объекты.
54.	В папке build проекта NetBeans находятся? >> хранятся скомпилированные файлы классов, имеющие расширение .class.
55.	В папке dist проекта NetBeans находятся? >> файлы, предназначенные для распространения как результат компиляции (модуль JAR приложения или библиотеки, а также документация к нему).
56.	В папке nbproject проекта NetBeans находятся? >> находится служебная информация по проекту.
57.	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.
58.	В папке test проекта NetBeans находятся? >> сопроводительные тесты, предназначенные для проверки правильности работы классов проекта.
59.	Команды документации (символы метаданных) @see? >> применяется для создания в документе гиперссылок на другие комментарии. Можно использовать для любых конструкций (классов, методов и т.д.).
60.	Команды документации (символы метаданных) @since? >> Информация о версии JDK, начиная с которой введён или работоспособен класс или интерфейс.
61.	Команды документации (символы метаданных) @throws? >> Информация об исключительных ситуациях, которые могут возбуждаться методом.
62.	Команды документации (символы метаданных) @param? >> информация о параметре метода.
63.	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.
64.	Целые типы, тип byte? >> Однобайтовое целое число (8-битное целое со знаком).
65.	Целые типы, тип short? >> Короткое целое число (16- битное целое со знаком).
66.	Целые типы, тип char? >> Символьный тип (беззнаковое 16- битное целое).
67.	Целые типы, тип int? >> Целое число (32- битное целое со знаком).

68.	Целые типы, тип long? >> Длинное целое число (64- битное целое со знаком).
69.	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
70.	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
71.	Вещественные типы, размер типа double? >> 8 байт
72.	Float f1 = new Float(Float.NaN); Float f2 = new Float(Float.NaN); System.out.println(""+ (f1 == f2)+" "+f1.equals(f2)+ " "+(Float.NaN == Float.NaN)); Что будет выведено в результате выполнения данного куска кода? >> false true false
73.	int i = 0; i++; System.out.print(i); i = i++; System.out.println(i); Что выведет данный код? >> 11
74.	int i = 0, j = 5; tp: for (;;) { i++; for (;;) { if (i > --j) { breaktp; } } System.out.println("i = " + i + ", j = " + j);} Что произойдет во время компиляции и выполнения данного фрагмента кода? >> Ошибка компиляции.
75.	int i = 1; //1 i = -(10 + 2 + i); //2 ++i--; //3 System.out.println(i); В какой строке(-ах) кода содержится ошибка? >> 3.
76.	Integer i = new Integer("10"); if (i.toString().intern() == i.toString().intern()){ System.out.println("Равный"); } else { System.out.println("Не равный"); } } Результат выполнения программы? >> Выведется "Равный".

77.	<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>
78.	<pre>Integer i = newInteger("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>
79.	<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>
80.	<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>
81.	<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>
82.	<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>
83.	<pre>Boolean b = new Boolean("Hello"); System.out.println(b);</pre> <p>Какой результат выполнения данного кода? >> false</p>
84.	<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>
85.	<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>

86.	<pre>int i = 0; System.out.print(i++); System.out.println(i);</pre> <p>Результат кода? >> 01</p>
87.	<pre>int j = 0; System.out.print(++j); System.out.println(j);</pre> <p>Результат кода? >> 11</p>
88.	<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>
89.	<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>
90.	<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>
91.	<pre>int result = 12+2*5%3-15/4; String x=12-6+"Hello"+7+5; System.out.println(result+"-"+x);</pre> <p>Что будет выведено на консоль? >> 10-6Hello75</p>
92.	<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");</pre> <p>Что выведет этот код? >> s1 не равно s2</p>
93.	<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");</pre> <p>Что выведет этот код? >> s1 равно s2</p>

94.	<pre>String s1="Строка"; String s2="Строка"; if(s1==s2) System.out.println("s1 равно s2"); else System.out.println("s1 не равно s2");</pre> <p>Что выведет этот код? >> s1 равно s2</p>
95.	<pre>String s=""; s+= 1+2+3; System.out.println(s);</pre> <p>Что выведет этот код? >> 6</p>
96.	<pre>String s="Сумма =" +1+2+3; System.out.println(s);</pre> <p>Что выведет этот код? >> Сумма=123</p>
97.	<pre>String s=1+2+" не равно "+1+2; System.out.println(s);</pre> <p>Что выведет этот код? >> 3 не равно 12</p>
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</i>. <i>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</i>. <i>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</i>. <i>String separator?</i> >> Содержит строку с символом разделителя между именами файлов и файловых масок в операционной системе.</p>
105.	<p>Методы класса <i>File</i>. <i>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 <u>папкой</u>. 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 <u>файлом</u>. 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.getAbsoluteFile()</i> ?</p> <p>>> Создание нового файлового объекта по абсолютному пути, соответствующему текущему файловому объекту.</p>
122.	<p>Методы класса <i>File.list()</i> ?</p> <p>>> 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.</p>
123.	<p>Конструкторы?</p> <p>>> это методы, используемые для инициализации объектов в процессе их реализации.</p>
124.	<p>Как объекты обмениваются информацией?</p> <p>>> Объекты общаются друг с другом посредством посылки сообщений.</p>
125.	<p>Что такое this?</p> <p>>> ссылка, с помощью которой объект обращается к самому себе.</p>
126.	<p>Класс ?</p> <p>>> Класс – это структурный тип данных, который включает описание полей данных, а также процедур и функций, работающих с этими полями данных.</p>
127.	<p>Процедуры и функции, работающие с полями данных класса?</p> <p>>> Называются методами.</p>
128.	<p>Объединение полей данных и процедур и функций, работающих с этими данными, в единый пакет, при наличии специальных правил доступа к элементам пакета?</p> <p>>> Инкапсуляция.</p>
129.	<p>Основные средства разработки классов. Наследование?</p> <p>>> Конструирование новых более сложных классов из уже имеющихся посредством добавления полей и определения новых методов (принцип иерархичности).</p>
130.	<p>Основные средства разработки классов. Полиморфизм?</p> <p>>> обеспечивает возможность задания различных реализаций некоторого единого по названию метода для классов различных уровней иерархии.</p>
131.	<p>Полиморфными объектами или полиморфными переменными называются?</p> <p>>> переменные, которым в процессе выполнения программы может быть присвоено значение, тип которого отличается от типа переменной.</p>

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.	int j = 0; for (int i = 0; i < 10; i++) j = j++; System.out.println(j); Что выведется в результате? >> 0
142.	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
143.	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
144.	. int x=0; for(int count=10; count < 5; count++){ x += count ;} Чему равна x? >> 0

145.	<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>
146.	<pre>class A {public A() {System.out.print("Создание класса A. ");}} class B extends A { public B(){System.out.print("Создание класса B. ");}} class C extends B {public C(){System.out.print("Создание класса C. ");}} public class testClass { public static void main(String[] args) { C c = new C(); }} Что выведется в результате? >> Создание класса A. Создание класса B. Создание класса C.</pre>
147.	<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>
148.	<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>
149.	<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>
150.	<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) { int x=3; process(x); System.out.println(x); }} >> 3</pre>
151.	<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>

152.	<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>
153.	<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); }} >> 32</pre>
154.	<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(); }} >> 0</pre>
155.	<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); }} >> 5</pre>
156.	<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;}} } >> 2</pre>
157.	<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>
158.	<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>

159.	<p>Дан код:</p> <pre>class Quest3 { public static void main(String s[]) { String args; System.out.print(args + s); }} </pre> <p>Результатом компиляции кода будет?</p> <p>>> ошибка компиляции: переменная args используется до инициализации.</p>
160.	<p>Дан код:</p> <pre>public class Quest5{ public static void main(){ System.out.print("А"); } public static void main(String args){ System.out.print("Б"); } public static void main(String[] args){ System.out.print("В");}} </pre> <p>Что будет выведено в результате компиляции и запуска?</p> <p>>> В</p>
161.	<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>>> а</p>
162.	<p>Дан код:</p> <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>В результате при компиляции и запуске будет выведено?</p> <p>>> 24</p>
163.	<p>Какие из объявлений корректны, если</p> <pre>class Owner{ class Inner{ }}? </pre> <p>>> new Owner().new Inner();</p>
164.	<p>Дан код:</p> <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>В результате при компиляции запуске будет выведено?</p> <p>>></p>
165.	<p>Экземпляром какого класса является поле <i>System.in</i>?</p> <p>>> java.io.InputStream;</p>
166.	<p>Какой абстрактный класс является суперклассом для всех классов, используемых для чтения байтов?</p> <p>>> InputStream</p>

167.	<p>Какой интерфейс наиболее пригоден для создания класса, содержащего несортированные уникальные объекты?</p> <p>>> Set</p>
168.	<p>Дан код:</p> <pre><applet code=MyApplet.class width=200 height=200> <param name=count value=5> </applet></pre> <p>Какой код читает параметр count в переменную i?</p> <p>>> int i = new Integer(getParameter("count")).intValue()</p> <p>Источник: http://yznaika.com/notes/629-java-answers</p>
169.	<p>Что произойдет с настройки безопасности (пользователи, роли, права), если резервная копия базы данных будет восстановлена на другом сервере?</p> <ul style="list-style-type: none"> o Будут потеряны все пользователи и все права, роли сохранятся без прав o База данных будет вообще непригодна к использованию o Будут потеряны все настройки безопасности • Все настройки сохраняться, но для пользователей может потребоваться восстановить логины o Будут потеряны все пользователи, роли и права сохранятся
170.	<p>Какое действие следует предпринять, чтобы минимизировать вероятность конфликта блокировок?</p> <ul style="list-style-type: none"> o Убедиться, что все транзакции изменяют данные в одной и той же последовательности o Гарантировать, что во всех транзакций выполняется только модификация данных • Использовать команду BEGIN DISTRIBUTED TRANSACTION всякий раз, когда Вам потребуется изменять данные в таблицах <ul style="list-style-type: none"> o Установить низкий приоритет мертвых блокировок o Гарантировать, что все таблицы имеют индексы для первичного и внешнего ключа
171.	<p>Какой из приведенных фрагментов кода позволит объявить курсор с характеристиками, необходимыми для решения поставленной задачи?</p> <ul style="list-style-type: none"> o DECLARE Prv CURSOR STATIC FOR SELECT cod_1,cod_2,summa FROM provodki ORDER BY cod_1 o DECLARE Prv CURSOR KEYSET FOR SELECT cod_1,cod_2,summa FROM provodki ORDER BY cod_1 • DECLARE Prv CURSOR DYNAMIC FOR SELECT cod_1,cod_2,summa FROM provodki ORDER BY cod_1 o DECLARE Prv CURSOR FAST_FORWARD FOR SELECT cod_1,cod_2,summa FROM provodki ORDER BY cod_1 o DECLARE Prv CURSOR FOR SELECT cod_1,cod_2,summa FROM provodki ORDER BY cod_1
172.	<p>В какой базе данных хранятся функции, определяемые пользователем?</p> <ul style="list-style-type: none"> o master o tempdb • В пользовательской o model o msdb
173.	<p>В какой из баз данных MS SQL Server создаются временные таблицы?</p> <ul style="list-style-type: none"> o master • В текущей o tempdb o msdb o model
174.	<p>Вы разработали хранимую процедуру, предназначенную для выполнения сложной выборки из нескольких связанных таблиц. При выполнении нагрузочного тестирования выясняется, что время отработки запроса слишком велико, и Вы решаете проанализировать план выполнения запроса.</p> <p>Какой из перечисленных клиентских инструментов позволит решить поставленную задачу?</p>

	<ul style="list-style-type: none"> o Enterprise Manager <ul style="list-style-type: none"> o Bcp o Query Analyzer <ul style="list-style-type: none"> o Qsql o Profiler
175.	<p>Вы установили в системе MS SQL Server режим неявных транзакций.</p> <ul style="list-style-type: none"> o Каким будет результат, если во время выполнения какого-либо запроса будет выдана явно команда BEGIN TRAN? o Будет выдано сообщение об ошибке, незавершенные транзакции откачены и выполнение запроса прервется <ul style="list-style-type: none"> • MS SQL Server будет автоматически остановлен, а затем перезапущен в режиме явных транзакций o Система будет переключена в режим явных транзакций без выдачи каких-либо сообщений и продолжит работу o Команда BEGIN TRAN будет проигнорирована, и система продолжит работу в режиме неявных транзакций o Будет сгенерировано и записано в лог предупредительное сообщение, и система продолжит работу в режиме неявных транзакций
176.	<p>Владельцем какого из перечисленных объектов не может являться пользователь базы данных?</p> <ul style="list-style-type: none"> o Таблица o Хранимая процедура o Представление o Функция, определяемая пользователем o Триггер
177.	<p>Поле CustomerID присутствует как в столбце Customers, так и в таблице Orders.</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="border: 1px solid gray; padding: 5px; width: 200px;"> <p>Customers</p> <ul style="list-style-type: none"> CustomerID Company Name Address City State </div> <div style="border: 1px solid gray; padding: 5px; width: 200px;"> <p>Orders</p> <ul style="list-style-type: none"> OrderID ProductID OrderDate Quantity CustomerID </div> </div> <p>Каждая покупка покупателя заносится как запись в таблицу Orders. Значение поля CustomerID таблицы Orders берется из таблицы Customers.</p> <p>Какого типа отношения описываются данным примером?</p> <ul style="list-style-type: none"> o "один к одному" o "один к двум" o "многие к одному" o "многие ко многим" • "один ко многим"
178.	<p>Какая из перечисленных ниже функций НЕ является агрегатной?</p> <ul style="list-style-type: none"> o SUM o AVG o MIN • ABS o VAR
179.	<p>Возможно ли использование MS SQL-2000 Developer для эксплуатации корпоративных приложений «клиент-сервер»?</p> <ul style="list-style-type: none"> o Нет, поскольку редакция Developer не имеет в комплекте поставки соответствующего инструментария o Да, при условии, что разработка приложений также выполнена с использованием этой же редакции MS SQL o Да, поскольку данная редакция позволяет использовать весь функционал MS SQL-2000 o Да, при условии, что и разработка, и эксплуатация, выполняются на одной и

	<p>той же инсталляции MS SQL</p> <ul style="list-style-type: none"> o Нет, поскольку это противоречит условиям лицензионного соглашения на редакцию Developer
180.	<p>Какой из перечисленных типов данных является оптимальным для хранения различных физических констант?</p> <ul style="list-style-type: none"> • numeric <ul style="list-style-type: none"> o float o money o int o datetime
181.	<p>Пользователь "Иван" входит в роль "Менеджеры". Роль "Менеджеры" входит в роль "Продавцы". Роли "Менеджеры" командой "Grant All" разрешен полный доступ ко всем таблицам базы данных. Роли "Продавцы" командой "Deny" запрещено удаление из таблицы "Склад", другие права на таблицу не установлены.</p> <p>Какими правами на таблицу "Склад" будет обладать пользователь "Иван"?</p> <ul style="list-style-type: none"> • Всеми ("All") <ul style="list-style-type: none"> o Правами владельца таблицы o Только правом чтения "Select" o "Select", "Reference" и "Insert" o "Select", "Update", "Reference" и "Insert"
182.	<p>Какое выражение в триггере на обновление таблицы DOC позволит определить, что значение колонки NUM изменено?</p> <ul style="list-style-type: none"> o exists(Select Num From Inserted) o exists(Select NUM From Deleted Where NUM not in (select NUM From DOC)) <ul style="list-style-type: none"> o NUM in Columns_Updated() o UPDATE(NUM) o @@ROWCOUNT > 0
183.	<p>Вы разработали хранимую процедуру, использующую механизм динамических вызовов SQL.</p> <p>Какова особенность доступа к таблицам для таких процедур?</p> <ul style="list-style-type: none"> o Пользователь, выполняющий вызов такой процедуры, должен иметь соответствующие права на используемые в динамическом вызове объекты <ul style="list-style-type: none"> o Динамический SQL не может обращаться к системным таблицам o Динамический SQL не может обращаться к представлениям o Динамический SQL не может обращаться к временным таблицам o Пользователь, выполняющий вызов такой процедуры, должен быть членом ролей sysadmin или db_owner
184.	<p>Пользователь "Иван" входит в роль "Менеджеры", которой разрешен полный доступ к таблице "dbo.Games", а также установлены другие разрешения. Необходимо, чтобы пользователь "Иван" не имел доступа к таблице "dbo.Games", но при этом обладал всеми прочими полномочиями роли "Менеджеры".</p> <p>Какая из перечисленных команд позволит решить поставленную задачу без изменения полномочий других пользователей базы данных?</p> <ul style="list-style-type: none"> o Revoke ALL On dbo.Games To [Иван] o Execute sp_droprolemember ' Менеджеры', 'Иван' o Revoke ALL On dbo.Games To [Менеджеры] o Deny ALL On dbo.Games To [Менеджеры] • Deny ALL On dbo.Games To [Иван]
185.	<p>Вы разработали хранимую процедуру, использующую механизм динамических вызовов SQL.</p> <p>Какова особенность доступа к таблицам для таких процедур?</p> <ul style="list-style-type: none"> o Пользователь, выполняющий вызов такой процедуры, должен быть членом ролей sysadmin или db_owner o Пользователь, выполняющий вызов такой процедуры, должен иметь соответствующие права на используемые в динамическом вызове объекты <ul style="list-style-type: none"> o Динамический SQL не может обращаться к системным таблицам o Динамический SQL не может обращаться к временным таблицам

	о Динамический SQL не может обращаться к представлениям
--	---

3.2 Кейс-задания.

№ задания	Формулировка задачи
186.	<p>0. Банковская система. Можно создать несколько банков. В каждом банке может быть несколько клиентов. Клиенты банка могут быть физическими и юридическими лицами. У каждого клиента банка существует счет (только один). У банка также существует аналогичный параметр – сумма собственных средств банка. Клиенты банка могут проводить операции:</p> <ul style="list-style-type: none"> • положить деньги на счет; • снять деньги со счета; • сделать перевод суммы денег на другой счет (возможно в другом банке, только для юридических лиц). <p>При переводе денег со счета на счет с суммы перевода в пользу банка взимается определенный процент. Необходимо предусмотреть интерфейс для создания/удаления банков, создания/удаления клиентов каждого банка, а также выполнения каждым клиентом перечисленных операций. При этом не должны допускаться операции, в результате которых счет клиента становится отрицательным.</p>
187.	<p>Маршрутное такси. Маршрутное такси движется по маршруту, который представляет собой остановки, соединенные прямыми линиями. Такси движется от начальной остановки к конечной и обратно и так множество раз. Такси имеет определенную вместимость. На остановках находятся пассажиры (экземпляры классов), которым надо попасть на определенную остановку. Такси останавливается на остановках, если есть свободные места или пассажиры, которым на этой остановке надо выходить и которые, соответственно, выходят. Пассажиры, которые стоят на остановках, садятся в такси, если оно движется в нужном направлении, в порядке очередности их появления на остановке. Задача заключается в моделировании движения такси. Необходимо предусмотреть возможность построения маршрута и добавления/удаления пассажиров на каждую остановку с заданием целевой остановки для каждого пассажира.</p>
188.	<p>Файловая система. Необходимо смоделировать файловую систему ОС Unix (без прав доступа). Файловая система состоит из папок, файлов и ссылок. Ссылки бывают 2-х видов жесткие и символичные (при удалении жесткой ссылки удаляется заодно и сам файл/папка, на который эта ссылка ссылается, точнее оба имени являются абсолютно равноправными). Файлы имеют размер, расширение, кроме того файлы и папки имеют атрибуты: время создания, время изменения и т.п. Файловая система имеет только один корень, представляющий собой папку, которую нельзя удалить. Для каждого элемента файловой системы предусмотреть операции создания, переименования, перемещения, изменения (меняется дата изменения) и удаления.</p>
189.	<p>Иерархия графических компонентов. Имеется набор компонентов: окно, надпись, список, кнопка, панель, линия. Компоненты окно и панель могут содержать другие компоненты. Компоненты могут быть видимы или скрыты, компоненты можно перемещать в пределах контейнера. Отобразить иерархию компонентов и навигацию по ней в виде дерева.</p>

190.	<p>Муравейник. Типы объектов: матка, солдаты, полицейские, рабочие, личинки, вредители. Каждый представитель является отдельным экземпляром класса. Муравейник имеет склад пищи фиксированного размера. Матка создает личинки, из личинок выводятся солдаты, полицейские или рабочие муравьи и ест очень много, солдаты уменьшают число вредителей и много едят, полицейские помогают увеличить число добываемой пищи и едят меньше солдат, рабочие добывают пищу и едят меньше полицейских, личинки только едят. Матка имеет следующие параметры:</p> <ul style="list-style-type: none"> • кол-во потребляемой пищи; • кол-во рождаемых личинок. <p>Солдаты имеют следующие параметры</p> <ul style="list-style-type: none"> • кол-во потребляемой пищи; • кол-во уничтожаемых вредителей. <p>Полицейские имеют следующие параметры:</p> <ul style="list-style-type: none"> • кол-во потребляемой пищи; • процент, на который увеличивается запас пищи. <p>Рабочие имеют следующие параметры:</p> <ul style="list-style-type: none"> • кол-во потребляемой пищи; • кол-во добываемой пищи. <p>Личинки имеют только один параметр: кол-во потребляемой пищи. Через некоторое время личинки должны разродиться в новую форму. При изменении формы личинки должны об этом сигнализировать матку, чтобы она определила тип, в который превратится личинка. Показать работу муравейника для различных значений числа муравьев, вредителей, пищи.</p>
191.	Учет результатов сдачи вступительных экзаменов
192.	Учет вкладов в банке
193.	Добровольное медицинское страхование
194.	Штатное расписание
195.	Учет затрат на изготовление изделий
196.	Учет эксплуатации транспортных средств
197.	Учет подписки на печатные издания
198.	Туристическое агентство
199.	Учет нарушений правил дорожного движения
200.	Учет сделок с недвижимостью
201.	Учет выполнения заказов
202.	Оптовый магазин. Учет продаж товара
203.	Прокат автомобилей
204.	Учет выполнения работ
205.	Платный прием в поликлинике
206.	Книжный магазин
207.	Чрезвычайные ситуации
208.	Отдел кадров
209.	Реализация товаров

210.	Библиотека
211.	Автовокзал

3.3 Вопросы к зачету

№	Текст вопроса
212.	Основные достоинства языка Java.
213.	Основные недостатки. Основные особенности.
214.	Версии Java.
215.	Апплеты.
216.	Жизненный цикл программы на Java.
217.	Структура пакета SDK.
218.	ООП в Java, наследование.
219.	Объектноориентированный язык программирования.
220.	5 принципов объектноориентированного подхода.
221.	Реализация принципов объектно-ориентированного подхода в Java.
222.	Операции в Java.
223.	Операции (operators) в языке Java.
224.	Литералы (константы).
225.	Арифметические. Логические литералы.
226.	Строковые литералы.
227.	Символьные литералы.
228.	Операторы (statements).
229.	Оператор - выражение.
230.	Условный оператор (if).
231.	Оператор return.
232.	Оператор цикла по предусловию (while).
233.	Оператор цикла по постусловию (do while).
234.	Оператор цикла "со счетчиком" (for).
235.	Операторы break и continue.
236.	Оператор выбора (switch).
237.	Массивы в Java.
238.	Массивы объектов.
239.	Многомерные массивы.
240.	Присваивание и копирование.
241.	Резюмируем основные правила.
242.	Конструкторы классов.
243.	Вызов одного конструктора из другого.
244.	Работа со строками (класс String).
245.	Структура Java программ.
246.	Знакомство с библиотеками и пакетами.
247.	Использование пакетов в программах.
248.	Создание своих собственных пакетов.
249.	Коллекции элементов.
250.	Статические вложенные классы.
251.	Коллекции объектов Java.
252.	Коллекции - это наборы произвольных объектов.
253.	Итераторы. Классы реализации коллекций.
254.	Коллекции- списки (List).
255.	Исключительные ситуации.
256.	Обработка исключительных ситуаций (Exceptions).
257.	Поведение программы при возникновении исключения.
258.	Структура и использование блока перехвата исключений.
259.	Классы исключительных ситуаций. Механизм контроля перехвата исключений.

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
ПКв-3 способностью использовать технологии разработки объектов профессиональной деятельности в областях: управление технологическими процессами, химическая промышленность, пищевая промышленность, а также предприятия различного профиля и все виды деятельности в условиях экономики информационного общества					
УМЕТЬ: умеет критически оценивать достоинства и недостатки, а также сильные и слабые стороны своей профессиональной деятельности	Собеседование (экзамен)	Критические оценивает свои достоинства и недостатки, понимает необходимость саморазвития и повышения квалификации в профессиональной деятельности	Обучающийся полностью раскрыл содержание материала в объеме, предусмотренном программой, изложил материал грамотным языком в определенной логической последовательности	зачтено	освоена (повышенный)
			Обучающийся твердо знает материал, грамотно и по существу излагает его, но допускает в ответе некоторые неточности	зачтено	освоена (повышенный)
			Обучающийся неполно или непоследовательно раскрыл содержание материала, но показал общее понимание вопроса, недостаточно правильно формулировки базовых понятий	зачтено	освоена (базовый)
			Обучающийся не раскрыл содержание материала, допускает грубые ошибки в формулировках основных понятий дисциплины	не зачтено	не освоена (недостаточный)
			Обучающийся не раскрыл содержание материала, допускает грубые ошибки в формулировках основных понятий дисциплины	не зачтено	не освоена (недостаточный)
ЗНАТЬ: основные алгоритмы типовых численных методов решения математических задач	Тестовое задание	Результат тестирования	Набрано менее 50% правильных ответов	не зачтено	не освоена (недостаточный)
			Набрано 50% - 74,99% правильных ответов	зачтено	освоена (базовый)
			Набрано 75% - 89,99% правильных ответов	зачтено	освоена (повышенный)
			Набрано 90% - 100% правильных ответов	зачтено	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	зачтено	освоена (повышенный)
			Обучающийся твердо знает материал, грамотно и по существу излагает его, но допускает в ответе некоторые неточ-	зачтено	освоена (повышенный)

			ности		
			Обучающийся не полностью или непоследовательно раскрыл содержание материала, но показал общее понимание вопроса, недостаточно правильные формулировки базовых понятий	зачтено	освоена (базовый)
			Обучающийся не раскрыл содержание материала, допускает грубые ошибки в формулировках основных понятий дисциплины	не зачтено	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 пользуется 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)