

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

УТВЕРЖДАЮ
Проректор по учебной работе

_____ Василенко В.Н.

«25» мая 2023 г.

РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ СИСТЕМЫ ПРОГРАММИРОВАНИЯ

Направление подготовки

09.03.02 Информационные системы и технологии

Направленность (профиль) подготовки

Разработка информационных систем и технологий

Квалификация выпускника

Бакалавр

Воронеж

1. Цели и задачи дисциплины

Целью дисциплины «Объектно-ориентированные системы программирования» является формирование компетенций обучающегося в области профессиональной деятельности и сфере профессиональной деятельности:

06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации средств вычислительной техники и информационных систем, управления их жизненным циклом);

40 Сквозные виды профессиональные деятельности в промышленности.

(в сфере организации и проведения научно-исследовательских и опытно-конструкторских работ в области информатики и вычислительной техники).

Дисциплина направлена на решение задач профессиональной деятельности следующих типов:

- производственно-технологический;
- проектный;
- 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, утвержденного приказом Министерства образования и науки Российской Федерации от 19.09.2017г. № 926.

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

№ п/п	Компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	УК-1	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
2	ПКв-3	ПКв-3 Способность интеграции программных модулей и компонент и верификации выпуска программных продуктов	ИД1 _{ПКв-3} – Проводит разработку процедур интеграции программных модулей ИД2 _{ПКв-3} – Осуществляет интеграцию программных модулей и компонент и верификацию выпусков программного продукта

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: основные теоретико-методологические положения системного подхода любой категории информации
	Умеет: осуществлять поиск информации для решения поставленной задачи по различным типам запросов и сопоставлять разные источники информации с целью выявления их противоречий и поиска достоверных суждений
	Владеет: методами анализа информации и предлагает возможные варианты решения поставленной задачи, оценивая их достоинства и недостатки
ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений	Знает: принципы сбора, отбора и обобщения информации
	Умеет: соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности
	Владеет: навыками работы с информационными источниками, опыт научного поиска, создания научных текстов

ИД1 _{ПКв-3} – Проводит разработку процедур интеграции программных модулей	Знает: знает регламент проведения разработки процедур интеграции программных модулей
	Умеет: проводить разработку процедур интеграции программных модулей
	Владеет: методами и алгоритмами разработки процедур интеграции программных модулей
ИД2 _{ПКв-3} – Осуществляет интеграцию программных модулей и компонент и верификацию выпусков программного продукта	Знает: основные методы интеграции программных модулей и компонент и алгоритмы верификации выпусков программного продукта
	Умеет: осуществлять интеграцию программных модулей и компонент и верификацию выпусков программного продукта
	Владеет: методами и алгоритмами интеграции программных модулей и компонент и верификации выпусков программного продукта

3. Место дисциплины в структуре ООП ВО

Дисциплина «Объектно-ориентированные системы программирования» относится к Части, формируемой участниками образовательных отношений Блока 1 ООП, модуль «Программирование». Дисциплина является обязательной к изучению.

Изучение дисциплины основано на знаниях, умениях и навыках, полученных обучающимися при изучении дисциплин предметной области по направлению подготовки бакалавров.

Дисциплина основана на данных следующих дисциплин: Основы программирования, Алгоритмы и структуры данных.

4. Объем дисциплины и виды учебных занятий

Общая трудоемкость дисциплины составляет 7 зачетных единицы.

Виды учебной работы	Всего ак. ч.	Распределение трудоемкости по семестрам, ак. ч.	
		4 семестр	5 семестр
Общая трудоемкость дисциплины (модуля)	252	180	72
Контактная работа в т. ч. аудиторные занятия:	104,45	57,1	47,35
Лекции	33	18	15
Практические занятия	33	36	30
Консультации текущие	1,65	0,9	0,75
Консультирование КР	1,5		1,5
Консультация перед экзаменом	2	2	-
Вид аттестации (зачет, экзамен)	0,3	0,2	0,1
Самостоятельная работа:	113,75	89,1	24,65
Проработка материалов по лекциям, учебникам, учебным пособиям	33	18	5
Подготовка к практическим занятиям	22	36	7
Курсовая работа	10		10
Кейс-задание	48,75	35,1	2,65
Подготовка к экзамену, зачет – контроль	33,8	33,8	-

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

5.1 Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела	Трудоемкость, ак. ч.
-------	---------------------------------	--------------------	----------------------

4 семестр			
1	Тема 1. Введение. Методология разработки объектно-ориентированного программного обеспечения	Введение в объектно-ориентированные методологии (технологии) разработки программных систем. Инструментальные средства, поддерживающие эти технологии ООП	23
2	Тема 2. Основные понятия и терминология объектно-ориентированного анализа и проектирования	Объектно-ориентированный анализ и проектирование: основные понятия и терминология. Цели анализа и проектирования. Сопоставление ОО языков программирования C# и Java	23
3	Тема 3. Инкапсуляция - центральное понятие объектно-ориентированного программирования	Основные принципы инкапсуляции. Реализация инкапсуляции в ОО языках программирования C# и Java	25
4	Тема 4. Наследование - базовое понятие объектно-ориентированного программирования	Основные принципы и правила наследования. Реализация наследования в ОО языках программирования C# и Java	23
5	Тема 5. Полиморфизм - базовое понятие в парадигме объектно-ориентированного программирования	Основные принципы и правила полиморфизма. Реализация наследования в ОО языках программирования C# и Java	25
6	Тема 6. Основы UML унифицированного языка моделирования объектно-ориентированных систем	Основы UML. Основные типы связей языка UML. Диаграммы вариантов использования. Язык UML. Основные элементы языка UML.	24,1
		Консультации текущие	0,9
		Консультации перед экзаменом	2
		Вид аттестации - экзамен	0,2
		Экзамен - контроль	33,8
5 семестр			
7	Тема 7. Основы объектно-ориентированного анализа	Понятие объектно-ориентированного анализа. Диаграммы классов. Диаграммы взаимодействия. Диаграммы состояний.	21,65
8	Тема 8. Объектно-ориентированный подход к созданию пользовательского интерфейса	Основные принципы объектно-ориентированного подхода к созданию пользовательского интерфейса. Реализация в ОО языках программирования C# и Java	17
9	Тема 9. Основы объектно-ориентированного проектирования	Принципы объектно-ориентированного проектирования программ. Реализация в ОО языках программирования C# и Java	14
10	Тема 10. Разработка компьютерных моделей реальных и концептуальных систем на основе методологии компонентно-ориентированного программирования	Диаграммы пакетов, компонентов и размещения. Разработка моделей реальных и концептуальных систем на основе методологии компонентно-ориентированного программирования в в ОО языках программирования C# и Java	17
		Консультации текущие	0,75
		Консультации КР	1,5
		Вид аттестации – зачет	0,1

5.2 Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции, ак. ч.	Практические занятия, ак. ч.	СРО, ак. ч.
4 семестр				
1	Тема 1. Введение. Методология разработки объектно-ориентированного программного обеспечения	4	4	15
2	Тема 2. Основные понятия и терминология объектно-	4	4	15

	ориентированного анализа и проектирования			
3	Тема 3. Инкапсуляция - центральное понятие объектно-ориентированного программирования	4	6	15
4	Тема 4. Наследование - базовое понятие объектно-ориентированного программирования	2	6	15
5	Тема 5. Полиморфизм - базовое понятие в парадигме объектно-ориентированного программирования	2	8	15
6	Тема 6. Основы UML унифицированного языка моделирования объектно-ориентированных систем	2	8	14,1
	Консультации текущие			0,9
	Консультации перед экзаменом			2
	Вид аттестации - экзамен			0,2
	Экзамен - контроль			33,8
5 семестр				
7	Тема 7. Основы объектно-ориентированного анализа	3	12	5,65
8	Тема 8. Объектно-ориентированный подход к созданию пользовательского интерфейса	3	8	6
9	Тема 9. Основы объектно-ориентированного проектирования	4	5	7
10	Тема 10. Разработка компьютерных моделей реальных и концептуальных систем на основе методологии компонентно-ориентированного программирования	5	5	6
	Консультации текущие			0,75
	Консультации КР			
	Вид аттестации – зачет			0,1

5.2.1 Лекции

№ п/п	Наименование раздела дисциплины	Тематика лекционных занятий	Трудоемкость, ак. ч.
4 семестр			
1	Тема 1. Введение. Методология разработки объектно-ориентированного программного обеспечения	Введение в объектно-ориентированные методологии (технологии) разработки программных систем. Инструментальные средства, поддерживающие эти технологии ООП	4
2	Тема 2. Основные понятия и терминология объектно-ориентированного анализа и проектирования	Объектно-ориентированный анализ и проектирование: основные понятия и терминология. Цели анализа и проектирования. Сопоставление ОО языков программирования С# и Java	4
3	Тема 3. Инкапсуляция - центральное понятие объектно-ориентированного программирования	Основные принципы инкапсуляции. Реализация инкапсуляции в ОО языках программирования С# и Java	4
4	Тема 4. Наследование - базовое понятие объектно-ориентированного программирования	Основные принципы и правила наследования. Реализация наследования в ОО языках программирования С# и Java	2
5	Тема 5. Полиморфизм - базовое понятие в парадигме объектно-ориентированного программирования	Основные принципы и правила полиморфизма. Реализация наследования в ОО языках программирования С# и Java	2
6	Тема 6. Основы UML унифицированного языка моделирования объектно-ориентированных систем	Основы UML. Основные типы связей языка UML. Диаграммы вариантов использования. Язык UML. Основные элементы языка UML.	2
5 семестр			
7	Тема 7. Основы объектно-ориентированного анализа	Понятие объектно-ориентированного анализа. Диаграммы классов. Диаграммы взаимодействия. Диаграммы состояний.	3
8	Тема 8. Объектно-ориентированный подход к	Основные принципы объектно-ориентированного подхода к созданию пользовательского интерфейса. Реализация в	3

	созданию пользовательского интерфейса	ОО языках программирования C# и Java	
9	Тема 9. Основы объектно-ориентированного проектирования	Принципы объектно-ориентированного проектирования программ. Реализация в ОО языках программирования C# и Java	4
10	Тема 10. Разработка компьютерных моделей реальных и концептуальных систем на основе методологии компонентно-ориентированного программирования	Диаграммы пакетов, компонентов и размещения. Разработка моделей реальных и концептуальных систем на основе методологии компонентно-ориентированного программирования в ОО языках программирования C# и Java	5

5.2.2 Практические занятия

№ п/п	Наименование раздела дисциплины	Тематика практических занятий	Трудоемкость, ак. ч.
4 семестр			
1	Тема 1. Введение. Методология разработки объектно-ориентированного программного обеспечения	Введение в объектно-ориентированные методологии (технологии) разработки программных систем. Инструментальные средства, поддерживающие эти технологии ООП	4
2	Тема 2. Основные понятия и терминология объектно-ориентированного анализа и проектирования	Объектно-ориентированный анализ и проектирование: основные понятия и терминология. Цели анализа и проектирования. Сопоставление ОО языков программирования C# и Java	4
3	Тема 3. Инкапсуляция - центральное понятие объектно-ориентированного программирования	Основные принципы инкапсуляции. Реализация инкапсуляции в ОО языках программирования C# и Java	6
4	Тема 4. Наследование - базовое понятие объектно-ориентированного программирования	Основные принципы и правила наследования. Реализация наследования в ОО языках программирования C# и Java	6
5	Тема 5. Полиморфизм - базовое понятие в парадигме объектно-ориентированного программирования	Основные принципы и правила полиморфизма. Реализация наследования в ОО языках программирования C# и Java	8
6	Тема 6. Основы UML унифицированного языка моделирования объектно-ориентированных систем	Основы UML. Основные типы связей языка UML. Диаграммы вариантов использования. Язык UML. Основные элементы языка UML.	8
5 семестр			
7	Тема 7. Основы объектно-ориентированного анализа	Понятие объектно-ориентированного анализа. Диаграммы классов. Диаграммы взаимодействия. Диаграммы состояний.	12
8	Тема 8. Объектно-ориентированный подход к созданию пользовательского интерфейса	Основные принципы объектно-ориентированного подхода к созданию пользовательского интерфейса. Реализация в ОО языках программирования C# и Java	8
9	Тема 9. Основы объектно-ориентированного проектирования	Принципы объектно-ориентированного проектирования программ. Реализация в ОО языках программирования C# и Java	5
10	Тема 10. Разработка компьютерных моделей реальных и концептуальных систем на основе методологии компонентно-ориентированного программирования	Диаграммы пакетов, компонентов и размещения. Разработка моделей реальных и концептуальных систем на основе методологии компонентно-ориентированного программирования в ОО языках программирования C# и Java	5

5.2.3 Лабораторные занятия - Не предусмотрены учебным планом

Проработка материалов по лекциям, учебникам, учебным пособиям	33	18	5
Подготовка к практическим занятиям	22	36	7
Курсовая работа	10		10
Кейс-задание	48,75	35,1	2,65

5.2.4 Самостоятельная работа обучающихся (СРО)

№ п/п	Наименование раздела дисциплины	Вид СРО	Трудоемкость, ак. ч.
4 семестр			
1	Тема 1. Введение. Методология разработки объектно-ориентированного программного обеспечения	Проработка материалов по лекциям, учебникам, учебным пособиям	3
		Подготовка к практическим занятиям	6
		Кейс-задание	6
2	Тема 2. Основные понятия и терминология объектно-ориентированного анализа и проектирования	Проработка материалов по лекциям, учебникам, учебным пособиям	3
		Подготовка к практическим занятиям	6
		Кейс-задание	6
3	Тема 3. Инкапсуляция - центральное понятие объектно-ориентированного программирования	Проработка материалов по лекциям, учебникам, учебным пособиям	3
		Подготовка к практическим занятиям	6
		Кейс-задание	6
4	Тема 4. Наследование - базовое понятие объектно-ориентированного программирования	Проработка материалов по лекциям, учебникам, учебным пособиям	3
		Подготовка к практическим занятиям	6
		Кейс-задание	6
5	Тема 5. Полиморфизм - базовое понятие в парадигме объектно-ориентированного программирования	Проработка материалов по лекциям, учебникам, учебным пособиям	3
		Подготовка к практическим занятиям	6
		Кейс-задание	6
6	Тема 6. Основы UML унифицированного языка моделирования объектно-ориентированных систем	Проработка материалов по лекциям, учебникам, учебным пособиям	3
		Подготовка к практическим занятиям	6
		2 Кейс-задание	5,1
5 семестр			
7	Тема 7. Основы объектно-ориентированного анализа	Проработка материалов по лекциям, учебникам, учебным пособиям	1
		Подготовка к практическим занятиям	2
		Кейс-задание	2,65
8	Тема 8. Объектно-ориентированный подход к созданию пользовательского интерфейса	Проработка материалов по лекциям, учебникам, учебным пособиям	1
		Подготовка к практическим занятиям	2
		КР	3
9	Тема 9. Основы объектно-ориентированного проектирования	Проработка материалов по лекциям, учебникам, учебным пособиям	2
		Подготовка к практическим занятиям	2
		КР	3
10	Тема 10. Разработка компьютерных моделей реальных и концептуальных систем на основе методологии компонентно-ориентированного программирования	Проработка материалов по лекциям, учебникам, учебным пособиям	1
		Подготовка к практическим занятиям	1
		КР	4

6 Учебно-методическое и информационное обеспечение дисциплины

Для освоения дисциплины обучающийся может использовать:

6.1 Основная литература

- 1 Конова, Е.А. Алгоритмы и программы. Язык С++: учеб. пособие. — Санкт-Петербург : Лань, 2017 <https://e.lanbook.com/reader/book/90158/#>
- 2 Тюкачев, Н.А. С#. Алгоритмы и структуры данных — Санкт-Петербург : Лань, 2017 <https://e.lanbook.com/reader/book/94748/#>
- 3 Тюкачев, Н.А. С#. Основы программирования . — Санкт-Петербург: Лань, 2017 <https://e.lanbook.com/reader/book/94749/#1>
- 4 Разработка программного обеспечения системы мониторинга производства на языке С++ с использованием математической модели технологического процесса: учеб. пособие / А.А. Хвостов [и др.].— Воронеж : ВГУИТ, 2014 <https://e.lanbook.com/book/72901>

6.2 Дополнительная литература:

- 1 Лаврищева, Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. — Москва : Издательство Юрайт, 2022. — 280 с. — (Высшее образование). — ISBN 978-5-534-01056-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/491048> (дата обращения: 24.02.2022).

6.3 Перечень учебно-методического обеспечения для самостоятельной работы студентов

1. Саввина Е.А., Объектно-ориентированные системы программирования [Электронный ресурс] : методические указания для СРО для студентов, обучающихся по направлению 09.03.01 – Информатика и вычислительная техника, очной и заочной форм обучения / Е.А. Саввина ; ВГУИТ, Кафедра корпоративных информационных систем и программирования. - Воронеж : ВГУИТ, 2022. - 32 с. - Электрон. ресурс. - <http://biblos.vsuet.ru/ProtectedView/Book/ViewBook/2581>

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

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
«Российское образование» - федеральный портал	http://www.edu.ru/index.php
Научная электронная библиотека	http://www.elibrary.ru/defaulttx.asp?
Федеральная университетская компьютерная сеть России	http://www.runnet.ru/
Информационная система «Единое окно доступа к образовательным ресурсам»	http://www.window.edu.ru/
Электронная библиотека ВГУИТ	http://biblos.vsuet.ru/megapro/web
Сайт Министерства науки и высшего образования РФ	http://minobrnauki.gow.ru
Портал открытого on-line образования	http://npoed.ru
Информационно-коммуникационные технологии в образовании. Система федеральных образовательных порталов	http://www.ict.edu.ru/
Электронная образовательная среда ФГБОУ ВО «ВГУИТ»	http://education.vsuet.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, современные профессиональные базы данных и информационные справочные системы: ЭИОС

университета, в том числе на базе программной платформы «Среда электронного обучения ЗКЛ», 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.

7 Материально-техническое обеспечение дисциплины (модуля)

<p>Аудитории для проведения занятий лекционного типа, лабораторных и практических занятий</p>	<p>Ауд. 420: Комплекты мебели для учебного процесса. ПЭВМ-12 (компьютер Core i5-4460), проектор Acer projector X1383WH, экран, стенды – 5 шт., блок управления комплекса радиоконтроля и поиска радиопередающих устройств «ОМЕГА» (переносной), МУ защиты ресурсов сети от внутренних и внешних атак CISCO ASA5505-KB, переносной комплекс для автоматизации измерений при проведении исследований и контроля технических средств ЭВТ «НАВИГА-ТОР-ПЗГ»; средство активной защиты информации от утечки за счет побочных электромагнитных излучений и наводок «СОНАТА-РЗ.1»; система защиты речевой информации «Соната-АВ-4Б» (Центральный блок питания и управления + Размыкатели в составе СВАЗ Соната АВ); профессиональный обнаружитель скрытых видеокамер СОКОЛ-М (переносной); портативный обнаружитель закладок Protect1203 (переносной); устройство активной защиты информации «ВЕТО-М»; электронный замок Samsung SHS-2920</p>	<p>Microsoft Windows 7 (64 разрядная) Профессиональная Лицензия (DreamSpark); Microsoft Office (standart) 2007 Профессиональная Лицензия (DreamSpark); Microsoft Access 2007 Профессиональная Лицензия (DreamSpark); Microsoft Project 2007 Профессиональная Лицензия (DreamSpark); Microsoft Share Point 2007 Профессиональная Лицензия (DreamSpark); Microsoft Visio 2007 Профессиональная Лицензия (DreamSpark) Microsoft SQL server 2008 Профессиональная Лицензия (DreamSpark); 1 С Предприятия Лицензия; 7-Zip File Manager (архиватор) Бесплатное ПО; Adobe Acrobat Reader (Бесплатное ПО); Adobe Flash Player (Бесплатное ПО); FAR file manager Бесплатное ПО; Google Chrome Бесплатное ПО; Java TM 7 (64-bit) Бесплатное ПО; K-Lite Codec Pack Бесплатное ПО; Mozilla Firefox Бесплатное ПО; Oracle VM VirtualBox Бесплатное ПО; Sublime Text Бесплатное ПО; Symantec Endpoint Protection 12 (Заменен на AVP Kaspersky) Бесплатное ПО; VMWare Player (Бесплатное ПО); Антивирус “Зоркий глаз” (Бесплатное ПО); Lazarus (аналог Delphi) Бесплатное ПО; SmathStudio (аналог Mathcad) Бесплатное ПО; NanoCAD (аналог Autocad) Бесплатное ПО; Gimp (графический редактор аналог Photoshop) Бесплатное ПО; Avidemax (видео редактор) Бесплатное ПО; Virtual Dub (видео редактор) Бесплатное ПО; Free Pascal (Бесплатное ПО); Страж NT вер.3.0 Сертификат ФСТЭК No 2145 30.07.2013 г.; Ревизор 1XP Сертификат ФСТЭК No 989 08.02.2015 г.; Ревизор 2XP Сертификат ФСТЭК No 990 08.02.2015 г.; Фикс 2.0.2 Сертификат ФСТЭК No1548 15.01.2015 г.; Ревизор сети вер.3.0 Сертификат ФСТЭК No3413 02.06.2015 г.; СЗИ Панцирь К Сертификат ФСТЭК No1973 09.12.2015 г.; СЗИ Dallas Lock 8.0 К Сертификат ФСТЭК No2720 25.09.2015; СЗИ Dallas Lock 8.0 С Сертификат ФСТЭК No2945 16.08.2013</p>
<p>Аудитории для проведения занятий лекционного типа, лабораторных и практических занятий</p>	<p>Ауд. 332а: Комплекты мебели для учебного процесса. ПЭВМ – 12 (компьютер Core i5-4570), средство активной защиты информации изделие «Салют 2000С» с регулятором выходного уровня шума, стенды – 5 шт. Ауд. 424: Комплекты мебели для учебного процесса. ПЭВМ – 12: рабочая станция CPU Core 2Duo E6300 – 1.86 – 10 шт, Celeron D2.8 – 2шт.; стенды – 3 Ауд. 420: Комплекты мебели для учебного процесса. ПЭВМ-12 (компьютер Core i5-4460),</p>	<p>Microsoft Windows 7 (64 разрядная) Профессиональная Лицензия (DreamSpark); Microsoft Windows 2003 Профессиональная Лицензия (DreamSpark); Microsoft Office (standart) 2007 Профессиональная Лицензия (DreamSpark); Microsoft Access 2007 Профессиональная Лицензия (DreamSpark); Microsoft Project 2007 Профессиональная Лицензия (DreamSpark); Microsoft Share Point 2007 Профессиональная Лицензия (DreamSpark); Microsoft Visio 2007 Профессиональная</p>

	<p>проектор Acer projector X1383WH, экран, стенды – 5 шт., блок управления комплекса радиоконтроля и поиска радиопередающих устройств «ОМЕГА» (переносной), МУ защиты ресурсов сети от внутренних и внешних атак CISCO ASA5505-KB, переносной комплекс для автоматизации измерений при проведении исследований и контроля технических средств ЭВТ «НАВИГАТОР-ПЗГ»; средство активной защиты информации от утечки за счет побочных электромагнитных излучений и наводок «СОНАТА-РЗ.1»; система защиты речевой информации «Соната-АВ-4Б» (Центральный блок питания и управления + Размыкатели в составе СВАЗ Соната АВ); профессиональный обнаружитель скрытых видеокамер СОКОЛ-М (переносной); портативный обнаружитель закладок Protect1203 (переносной); устройство активной защиты информации «ВЕТО-М»; электронный замок Samsung SHS-2920</p>	<p>Лицензия (DreamSpark) Microsoft SQL server 2008 Профессиональная Лицензия (DreamSpark); 1 С Предприятие Лицензия; 7-Zip File Manager (архиватор) Бесплатное ПО; Adobe Acrobat Reader Бесплатное ПО; Adobe Flash Player Бесплатное ПО; FAR file manager Бесплатное ПО; Google Chrome Бесплатное ПО; Java TM 7 (64-bit) Бесплатное ПО; K-Lite Codec Pack Бесплатное ПО; Mozilla Firefox Бесплатное ПО; Oracle VM VirtualBox Бесплатное ПО; Sublime Text Бесплатное ПО; Symantec Endpoint Protection 12 (Заменен на AVP Kaspersky) Бесплатное ПО; VMWare Player Бесплатное ПО; Антивирус “Зоркий глаз” Бесплатное ПО; Lazarus (аналог Delphi) Бесплатное ПО; Smath Studio (аналог Mathcad) Бесплатное ПО; NanoCAD (аналог Autocad) Бесплатное ПО; Gimp (графический редактор аналог Photoshop) Бесплатное ПО; Avidemux (видео редактор) Бесплатное ПО; Virtual Dub (видео редактор) Бесплатное ПО; Free Pascal Бесплатное ПО (ауд.420) Страж NT вер.3.0 Сертификат ФСТЭК No 2145 30.07.2013 г.; Ревизор 1XP Сертификат ФСТЭК No 989 08.02.2015 г.; Ревизор 2XP Сертификат ФСТЭК No 990 08.02.2015 г.; Фикс 2.0.2 Сертификат ФСТЭК No1548 15.01.2015 г.; Ревизор сети вер.3.0 Сертификат ФСТЭК No3413 02.06.2015 г.; СЗИ Панцирь К Сертификат ФСТЭК No1973 09.12.2015 г.; СЗИ Dallas Lock 8.0 К Сертификат ФСТЭК No2720 25.09.2015; СЗИ Dallas Lock 8.0 С Сертификат ФСТЭК No2945 16.08.2013</p>
<p>Аудитории для самостоятельной работы, курсового и дипломного проектирования</p>	<p>Читальные залы библиотеки: Компьютеры со свободным доступом в сеть Интернет и Электронными библиотечными и информационно справочными системами; Ауд.424: Комплекты мебели для учебного процесса. Количество ПЭВМ – 12 (рабочая станция CPU Core 2Duo E6300 – 1.86 – 10 шт, Celeron D2.8 – 2 шт.), стенды – 3</p>	

8 Оценочные материалы для промежуточной аттестации обучающихся по дисциплине

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.**

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

ПРИЛОЖЕНИЕ к рабочей программе

1. Организационно-методические данные дисциплины для заочной форм обучения

1.1 Объемы различных форм учебной работы и виды контроля в соответствии с учебным планом (заочная форма)

Общая трудоёмкость дисциплины составляет 7 зачётных единицы.

Виды учебной работы	Всего ак. ч.	Распределение трудоёмкости по семестрам, ак. ч.	
		3 семестр	4 семестр
Общая трудоёмкость дисциплины (модуля)	252	180	72
Контактная работа в т.ч. аудиторные занятия:	33,2		
Лекции	12	6	6
Практические занятия (ПЗ)	14	6	8
Консультации текущие	2,1	0,9	1,2
Консультации КР	2		2
Консультация перед экзаменом	2	2	
Вид аттестации (экзамен, зачет)	0,3	0,2	0,1
Самостоятельная работа:	208,1	157,3	50,8
Проработка материалов по учебной литературе	162,1	139,3	22,8
Изучение материалов, изложенных в лекциях	8	4	4
Подготовка отчета к защите по практическим занятиям КР	8	4	4
КР	10		10
Контрольная работа	20	10	10
Подготовка к экзамену	10,7	6,8	3,9

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

по дисциплине

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ СИСТЕМЫ ПРОГРАММИРОВАНИЯ

1 Перечень компетенций с указанием этапов их формирования

№ п/п	Компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	УК-1	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
2	ПКв-3	ПКв-3 Способность интеграции программных модулей и компонент и верификации выпуска программных продуктов	ИД1 _{ПКв-3} – Проводит разработку процедур интеграции программных модулей ИД2 _{ПКв-3} – Осуществляет интеграцию программных модулей и компонент и верификацию выпусков программного продукта

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: основные теоретико-методологические положения системного подхода любой категории информации
	Умеет: осуществлять поиск информации для решения поставленной задачи по различным типам запросов и сопоставлять разные источники информации с целью выявления их противоречий и поиска достоверных суждений
	Владеет: методами анализа информации и предлагает возможные варианты решения поставленной задачи, оценивая их достоинства и недостатки
ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений	Знает: принципы сбора, отбора и обобщения информации
	Умеет: соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности
	Владеет: навыками работы с информационными источниками, опыт научного поиска, создания научных текстов
ИД1 _{ПКв-3} – Проводит разработку процедур интеграции программных модулей	Знает: знает регламент проведения разработки процедур интеграции программных модулей
	Умеет: проводить разработку процедур интеграции программных модулей
	Владеет: методами и алгоритмами разработки процедур интеграции программных модулей
ИД2 _{ПКв-3} – Осуществляет интеграцию программных модулей и компонент и верификацию выпусков программного продукта	Знает: основные методы интеграции программных модулей и компонент и алгоритмы верификации выпусков программного продукта
	Умеет: осуществлять интеграцию программных модулей и компонент и верификацию выпусков программного продукта
	Владеет: методами и алгоритмами интеграции программных модулей и компонент и верификации выпусков программного продукта

2 Паспорт оценочных материалов по дисциплине

№ п/п	Разделы дисциплины	Индекс контролируемой компетенции (или ее части)	Оценочные средства		Технология/процедура оценивания (способ контроля)
			Наименование	№№ заданий	
1	Введение. Методология	ПКв-3 УК-1	Тест	1-3, 16-17	Компьютерное тестирование Процентная шкала. 0-100 %;

	разработки объектно-ориентированного программного обеспечения				0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Собеседование (вопросы для экзамена)	42-44, 61-63	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
			Подгонка к практическим работам,	97	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Кейс-задание	81, 90	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
2	Основные понятия и терминология объектно-ориентированного анализа и проектирования	ПКв-3 УК-1	Тест	3-4, 17-19	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Собеседование (вопросы для экзамена)	45-48, 64-68	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
			Подгонка к практическим работам	98	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Кейс-задание	82, 91	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
3	Инкапсуляция - центральное понятие объектно-ориентированного программирования	ПКв-3 УК-1	Тест	5-6, 20-23	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Собеседование (вопросы для экзамена)	49-50, 69-70	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
			Подгонка к практическим работам	99	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Кейс-задание	83. 92	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
4	Наследование - базовое понятие объектно-	ПКв-3 УК-1	Тест	7-8, 24-25	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно;

	ориентированного программирования				75- 84,99% -хорошо; 85-100% - отлично.
			Собеседование (вопросы для экзамена)	51-53, 70-71	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
			Подгонка к практическим работам	100	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Кейс-задание	84, 93	Проверка преподавателем Отметка в системе «зачтено – не зачтено»

5	Полиморфизм - базовое понятие в парадигме объектно-ориентированного программирования	ПКв-3 УК-1	Тест	9, 26-28	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Собеседование (вопросы для экзамена)	54, 72-73	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
			Подгонка к практическим работам	101	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Кейс-задание	85, 94	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
6	Основы UML унифицированного языка моделирования объектно-ориентированных систем	ПКв-3 УК-1	Тест	10, 29-30	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Собеседование (вопросы для экзамена)	55, 74	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
			Подгонка к практическим работам	102	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Кейс-задание	86, 95	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
7	Основы объектно-ориентированного анализа	ПКв-3 УК-1	Тест	11, 31-33	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо;

					85-100% - отлично.
			Собеседование (вопросы для экзамена)	56, 76	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
			Подгонка к практическим работам	103	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Кейс-задание	87-89, 96	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
8	Объектно-ориентированный подход к созданию пользовательского интерфейса	ПКв-3 УК-1	Тест	12, 34-37	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Собеседование (вопросы для экзамена)	57,77	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
			Подгонка к практическим работам	104	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Курсовая работа	107 -111, 119-120	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
9	Основы объектно-ориентированного проектирования	ПКв-3 УК-1	Тест	13, 38-39	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Собеседование (вопросы для экзамена)	58, 78	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
			Подгонка к практическим работам	105	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
			Курсовая работа	112-115, 121-127	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
10	Разработка компьютерных моделей реальных и концептуальных систем на	ПКв-3 УК-1	Тест	14-15. 40-41	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.

основе методологии компонентно-ориентированного программирования	Собеседование (вопросы для экзамена)	59-60, 79-80	Проверка преподавателем Отметка в системе «зачтено – не зачтено»
	Подгонка к практическим работам	106	Компьютерное тестирование Процентная шкала. 0-100 %; 0-59,99% - неудовлетворительно; 60-74,99% - удовлетворительно; 75- 84,99% -хорошо; 85-100% - отлично.
	Курсовая работа	116-118, 128-134	Проверка преподавателем Отметка в системе «зачтено – не зачтено»

3 Оценочные материалы для промежуточной аттестации

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Для оценки знаний, умений, навыков студентов по дисциплине применяется бально-рейтинговая система оценки сформированности компетенций студента.

Бально-рейтинговая система оценки осуществляется в течение всего семестра при проведении аудиторных занятий и контроля самостоятельной работы. Показателями ОМ являются: текущий опрос в виде собеседования на лабораторных работах, тестовые задания и самостоятельно (домашнее задание). Оценки выставляются в соответствии с графиком контроля текущей успеваемости студентов в автоматизированную систему баз данных (АСУБД) «Рейтинг студентов».

Обучающийся, набравший в семестре более 60 % от максимально возможной бально-рейтинговой оценки работы в семестре получает зачет автоматически.

Студент, набравший за текущую работу в семестре менее 60 %, 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 зачет.

Аттестация обучающегося по дисциплине проводится в форме тестирования и предусматривает возможность последующего собеседования (экзамена). Зачет проводится в виде тестового задания.

Каждый вариант теста включает 30 контрольных заданий, из них:

- 10 контрольных заданий на проверку знаний;
- 10 контрольных заданий на проверку умений;
- 10 контрольных заданий на проверку навыков;

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

3.1 Тесты (тестовые задания и кейс-задания)

3.1.1 Шифр и наименование компетенции

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

№ задания	Тестовое задание
	Выбрать один ответ из 4 ответов

1.	<p>Что такое ООП?</p> <p>a) Объектно-ориентированное программирование b) Оконно-ориентированное программирование c) Облачно-ориентированное программирование d) Объединенное-ориентированное программирование</p>
2.	<p>Что такое класс?</p> <p>a) Экземпляр объекта b) Тип данных c) Шаблон для создания объектов d) Название переменной</p>
3.	<p>Что такое объект?</p> <p>a) Способ работы с классом b) Экземпляр класса c) Метод класса d) Переменная класса</p>
4.	<p>Какой из этих принципов ООП отвечает за связь между объектами?</p> <p>a) Наследование b) Инкапсуляция c) Абстракция d) Полиморфизм</p>
5.	<p>Что такое инкапсуляция?</p> <p>a) Способ скрыть реализацию объекта b) Способ создания нового класса на основе старого c) Способ создания экземпляра класса d) Способ создания метода класса</p>
6.	<p>Какой из этих принципов ООП отвечает за разделение ответственности между классами?</p> <p>a) Наследование b) Инкапсуляция c) Абстракция d) Принцип единственной ответственности</p>
7.	<p>Что такое наследование?</p> <p>a) Способ скрыть реализацию объекта b) Способ создания нового класса на основе старого c) Способ создания экземпляра класса d) Способ создания метода класса</p>
8.	<p>Что такое абстракция?</p> <p>a) Способ скрыть реализацию объекта b) Способ создания нового класса на основе старого c) Способ создания экземпляра класса d) Способ создания метода класса</p>
9.	<p>Что такое полиморфизм?</p> <p>a) Способ скрыть реализацию объекта b) Способ создания нового класса на основе старого c) Способ создания экземпляра класса d) Способ создания метода класса</p>
10.	<p>Что такое перегрузка методов?</p> <p>a) Возможность создать метод с одним и тем же названием, но разными параметрами b) Возможность создать метод с одним и тем же названием и параметрами c) Возможность изменить существующий метод в процессе выполнения программы d) Возможность создания методов вложенных в другие методы</p>
11.	<p>Что такое переопределение методов?</p> <p>a) Возможность изменить существующий метод в процессе выполнения программы b) Возможность создания методов вложенных в другие методы c) Возможность создать метод с одним и тем же названием и параметрами d) Возможность создать метод в наследуемом классе с тем же названием, что и в базовом классе, но с измененной реализацией</p>
12.	<p>Что такое интерфейс?</p> <p>a) Класс, который не может иметь реализации методов b) Класс, который может иметь только статические методы c) Класс, который может иметь только нестатические методы</p>

	d) Класс, который не может иметь переменных
13.	Что такое абстрактный класс? a) Класс, который не может иметь реализации методов b) Класс, который может иметь только статические методы c) Класс, который может иметь только нестатические методы d) Класс, который не может иметь переменных и может иметь только абстрактные методы
14.	Что такое конструктор? a) Метод, который вызывается при создании объекта класса b) Метод, который вызывается при удалении объекта класса c) Метод, который вызывается при изменении объекта класса d) Метод, который вызывается при создании класса
15.	Что такое деструктор? a) Метод, который вызывается при создании объекта класса b) Метод, который вызывается при удалении объекта класса c) Метод, который вызывается при изменении объекта класса d) Метод, который вызывается при создании класса

3.2.1 Шифр и наименование компетенции

ПКв-3 Способность интеграции программных модулей и компонент и верификации выпуска программных продуктов

№ задания	Тестовое задание
	Выбрать один ответ из 4 ответов
16.	Что такое this? a) Указатель на экземпляр объекта b) Указатель на класс c) Указатель на метод d) Указатель на переменную
17.	Что такое статические поля? a) Поля, которые принадлежат экземпляру объекта b) Поля, которые принадлежат классу c) Поля, которые принадлежат методу d) Поля, которые принадлежат переменной
18.	Что такое статические методы? a) Методы, которые принадлежат экземпляру объекта b) Методы, которые принадлежат классу c) Методы, которые принадлежат методу d) Методы, которые принадлежат переменной
19.	Что такое модификатор доступа public? a) Модификатор доступа, который позволяет получить доступ к полям и методам из любого места программы b) Модификатор доступа, который позволяет получить доступ к полям и методам только из класса, где они определены c) Модификатор доступа, который позволяет получить доступ к полям и методам только из классов, наследующих данный класс d) Модификатор доступа, который запрещает доступ к полям и методам из любого места программы
20.	Что такое модификатор доступа private? a) Модификатор доступа, который позволяет получить доступ к полям и методам из любого места программы b) Модификатор доступа, который позволяет получить доступ к полям и методам только из класса, где они определены c) Модификатор доступа, который позволяет получить доступ к полям и методам только из классов, наследующих данный класс d) Модификатор доступа, который запрещает доступ к полям и методам из любого места программы
21.	Что такое модификатор доступа protected? a) Модификатор доступа, который позволяет получить доступ к полям и методам из любого места программы b) Модификатор доступа, который позволяет получить доступ к полям и методам только из класса, где они определены c) Модификатор доступа, который позволяет получить доступ к полям и методам только из классов, наследующих данный класс

	d) Модификатор доступа, который запрещает доступ к полям и методам из любого места программы
22.	Что такое инкапсуляция данных? a) Способ скрыть реализацию объекта b) Способ скрыть доступ к полям и методам c) Способ создания экземпляра класса d) Способ создания метода класса
23.	Что такое инкапсуляция поведения? a) Способ скрыть реализацию объекта b) Способ скрыть доступ к полям и методам c) Способ создания экземпляра класса d) Способ скрыть реализацию методов
24.	Что такое композиция? a) Способ создания нового класса на основе старого b) Способ создания методов вложенных в другие методы c) Способ создания объекта на основе другого объекта d) Способ изменить существующий метод в процессе выполнения программы
25.	Что такое агрегация? a) Способ создания нового класса на основе старого b) Способ создания методов вложенных в другие методы c) Способ создания объекта, который содержит ссылки на другие объекты d) Способ изменить существующий метод в процессе выполнения программы
26.	Что такое полиморфизм параметров? a) Возможность создать метод с одним и тем же названием, но разными параметрами b) Возможность создать метод с одним и тем же названием и параметрами c) Возможность изменить существующий метод в процессе выполнения программы d) Возможность создания методов вложенных в другие методы
27.	Что такое полиморфизм подтипов? a) Возможность создать метод с одним и тем же названием, но разными параметрами b) Возможность создать метод с одним и тем же названием и параметрами c) Возможность изменить существующий метод в процессе выполнения программы d) Возможность использовать объекты наследников вместо объектов базового класса
28.	Что такое перегрузка операторов? a) Возможность создать метод с одним и тем же названием, но разными параметрами b) Возможность создать метод с одним и тем же названием и параметрами c) Возможность изменить существующий метод в процессе выполнения программы d) Возможность определить поведение операторов для пользовательских типов данных
29.	Что такое обобщенное программирование? a) Способ программирования, основанный на использовании объектов b) Способ программирования, основанный на использовании функций c) Способ программирования, который позволяет создавать универсальные алгоритмы для разных типов данных d) Способ программирования, который позволяет создавать универсальные интерфейсы для разных типов объектов
30.	Что такое шаблоны проектирования? a) Универсальные алгоритмы для разных типов данных b) Универсальные интерфейсы для разных типов объектов c) Универсальные решения для типичных проблем проектирования d) Универсальные решения для типичных проблем программирования
31.	Что такое шаблон Singleton? a) Шаблон проектирования, который гарантирует, что класс имеет только один экземпляр и обеспечивает глобальную точку доступа к этому экземпляру b) Шаблон проектирования, который используется для создания универсальных интерфейсов для разных типов объектов c) Шаблон проектирования, который используется для создания универсальных алгоритмов для разных типов данных d) Шаблон проектирования, который используется для создания универсальных решений для типичных проблем программирования
32.	Что такое шаблон Factory Method? a) Шаблон проектирования, который используется для создания универсальных интерфейсов для разных типов объектов b) Шаблон проектирования, который используется для создания универсальных

	<p>алгоритмов для разных типов данных</p> <p>с) Шаблон проектирования, который гарантирует, что класс имеет только один экземпляр и обеспечивает глобальную точку доступа к этому экземпляру</p> <p>d) Шаблон проектирования, который используется для создания объектов без явного указания их классов</p>
33.	<p>Что такое шаблон Builder?</p> <p>a) Шаблон проектирования, который используется для создания объектов без явного указания их классов</p> <p>b) Шаблон проектирования, который используется для создания универсальных алгоритмов для разных типов данных</p> <p>с) Шаблон проектирования, который используется для создания универсальных интерфейсов для разных типов объектов</p> <p>d) Шаблон проектирования, который используется для создания сложных объектов из простых компонентов</p>
34.	<p>Что такое шаблон Observer?</p> <p>a) Шаблон проектирования, который используется для создания сложных объектов из простых компонентов</p> <p>b) Шаблон проектирования, который используется для создания объектов без явного указания их классов</p> <p>с) Шаблон проектирования, который используется для создания универсальных алгоритмов для разных типов данных</p> <p>d) Шаблон проектирования, который используется для уведомления объектов об изменении состояния других объектов</p>
35.	<p>Что такое шаблон Strategy?</p> <p>a) Шаблон проектирования, который используется для уведомления объектов об изменении состояния других объектов</p> <p>b) Шаблон проектирования, который используется для создания сложных объектов из простых компонентов</p> <p>с) Шаблон проектирования, который используется для создания универсальных алгоритмов для разных типов данных</p> <p>d) Шаблон проектирования, который используется для определения семейства алгоритмов и инкапсуляции каждого из них, позволяя выбирать алгоритмы на лету в зависимости от ситуации</p>
36.	<p>Что такое шаблон Decorator?</p> <p>a) Шаблон проектирования, который используется для определения семейства алгоритмов и инкапсуляции каждого из них, позволяя выбирать алгоритмы на лету в зависимости от ситуации</p> <p>b) Шаблон проектирования, который используется для создания сложных объектов из простых компонентов</p> <p>с) Шаблон проектирования, который используется для создания универсальных интерфейсов для разных типов объектов</p> <p>d) Шаблон проектирования, который используется для создания объектов без явного указания их классов</p>
37.	<p>Что такое шаблон Proxy?</p> <p>a) Шаблон проектирования, который используется для создания объектов без явного указания их классов</p> <p>b) Шаблон проектирования, который используется для создания универсальных интерфейсов для разных типов объектов</p> <p>с) Шаблон проектирования, который используется для определения семейства алгоритмов и инкапсуляции каждого из них, позволяя выбирать алгоритмы на лету в зависимости от ситуации</p> <p>d) Шаблон проектирования, который используется для создания объекта-заместителя, который может контролировать доступ к другому объекту</p>
38.	<p>Что такое шаблон Adapter?</p> <p>a) Шаблон проектирования, который используется для создания объекта-заместителя, который может контролировать доступ к другому объекту</p> <p>b) Шаблон проектирования, который используется для создания объектов без явного указания их классов</p> <p>с) Шаблон проектирования, который используется для создания универсальных интерфейсов для разных типов объектов</p> <p>d) Шаблон проектирования, который используется для определения семейства алгоритмов и инкапсуляции каждого из них, позволяя выбирать алгоритмы на лету в зависимости от ситуации</p>

39.	<p>Что такое шаблон Template Method?</p> <p>a) Шаблон проектирования, который используется для создания универсальных интерфейсов для разных типов объектов</p> <p>b) Шаблон проектирования, который используется для создания объектов без явного указания их классов</p> <p>c) Шаблон проектирования, который используется для определения основного алгоритма в базовом классе и отделения деталей реализации в подклассах</p> <p>d) Шаблон проектирования, который используется для определения семейства алгоритмов и инкапсуляции каждого из них, позволяя выбирать алгоритмы на лету в зависимости от ситуации</p>
40.	<p>Что такое шаблон Iterator?</p> <p>a) Шаблон проектирования, который используется для создания универсальных интерфейсов для разных типов объектов</p> <p>b) Шаблон проектирования, который используется для создания объектов без явного указания их классов</p> <p>c) Шаблон проектирования, который используется для обхода элементов коллекции без раскрытия ее внутренней структуры</p> <p>d) Шаблон проектирования, который используется для определения семейства алгоритмов и инкапсуляции каждого из них, позволяя выбирать алгоритмы на лету в зависимости от ситуации</p>
41.	<p>Что такое this?</p> <p>a) Указатель на экземпляр объекта</p> <p>b) Указатель на класс</p> <p>c) Указатель на метод</p> <p>d) Указатель на переменную</p>

3.2 Собеседование (вопросы для экзамена)

3.2.1 Шифр и наименование компетенции

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

Номер вопроса	Текст вопроса
42.	<p>1.Что такое ООП и каковы некоторые из его ключевых понятий?</p> <p>Ответ: Объектно-ориентированное программирование (ООП) — это парадигма программирования, основанная на концепции «объектов», являющихся экземплярами классов. Некоторые ключевые концепции ООП включают инкапсуляцию, наследование и полиморфизм.</p>
43.	<p>2.Что такое инкапсуляция?</p> <p>Ответ: Инкапсуляция — это практика сокрытия внутренней работы объекта от внешнего мира, чтобы изменения внутреннего состояния объекта не влияли на другие части программы.</p>
44.	<p>3.Что такое наследование?</p> <p>Ответ: Наследование — это механизм, с помощью которого один класс может наследовать свойства и методы от другого класса. Класс, от которого происходит, называется «базовым классом» или «суперклассом», а класс, наследующий свойства и методы, называется «производным классом» или «подклассом».</p>
45.	<p>4.Что такое полиморфизм?</p> <p>Ответ: Полиморфизм — это способность объектов принимать разные формы или вести себя по-разному в зависимости от контекста, в котором они используются. Этого можно добиться с помощью наследования, интерфейсов или перегрузки функций.</p>
46.	<p>5.Что такое конструктор?</p> <p>Ответ: Конструктор — это специальный метод, который вызывается при создании объекта класса. Его цель — инициализировать состояние объекта, обычно путем установки значений его переменных экземпляра.</p>
47.	<p>6.Что такое интерфейс?</p> <p>Ответ. Интерфейс — это набор абстрактных методов, которые могут быть реализованы любым классом, реализующим интерфейс. Интерфейсы используются для определения</p>

	контракта, которого должен придерживаться класс, без указания того, как класс должен реализовывать методы.
48.	7.Что такое статический метод? Ответ. Статический метод — это метод, который принадлежит самому классу, а не какому-либо экземпляру класса. Статические методы можно вызывать непосредственно в классе, без необходимости сначала создавать экземпляр класса.
49.	8.Что такое переопределение метода? Ответ: Переопределение метода — это процесс, посредством которого подкласс предоставляет другую реализацию метода, который уже определен в его суперклассе. Это позволяет подклассу изменять поведение метода без изменения его сигнатуры.
50.	9.Что такое суперкласс? Ответ: Суперкласс — это класс, производный от другого класса. Суперкласс предоставляет набор свойств и методов, которые могут наследоваться его подклассами.
51.	10.Что такое подкласс? Ответ: Подкласс — это класс, производный от суперкласса. Подкласс наследует свойства и методы своего суперкласса, а также может добавлять собственные свойства и методы.
52.	11.Что такое наследование? Чем он отличается от композиции? Ответ: Наследование — это механизм, с помощью которого один класс приобретает свойства и методы другого класса. Композиция, с другой стороны, включает в себя создание объектов других классов внутри класса для использования их функциональных возможностей. Основное различие между ними заключается в том, что наследование представляет собой отношение «является» между классами, а композиция представляет собой отношение «имеет».
53.	12.Что такое абстракция в ООП? Ответ: Абстракция — это процесс выделения существенных признаков объекта и игнорирования второстепенных. В ООП абстракция достигается за счет использования абстрактных классов и интерфейсов.
54.	13.Что такое полиморфизм в ООП? Чем это отличается от перегрузки? Ответ: Полиморфизм — это способность объекта принимать различные формы. В ООП это достигается за счет наследования и интерфейсов. С другой стороны, перегрузка относится к процессу определения нескольких методов с одним и тем же именем, но с разными параметрами.
55.	14.Что такое абстрактный класс? Чем он отличается от конкретного класса? Ответ. Абстрактный класс — это класс, который не может быть создан и предназначен для создания подклассов. Он может содержать один или несколько абстрактных методов, которые объявлены, но не реализованы. С другой стороны, конкретный класс — это класс, который может быть создан и содержит все детали реализации его методов
56.	15.Что такое интерфейс? Чем он отличается от абстрактного класса? Ответ. Интерфейс — это набор сигнатур методов, которые могут быть реализованы любым классом, реализующим интерфейс. В отличие от абстрактного класса интерфейс не может содержать никаких деталей реализации, и все методы должны быть реализованы реализующим классом.
57.	16.Что такое конструктор? Чем он отличается от метода? Ответ: Конструктор — это специальный метод, который вызывается при создании объекта класса. Его цель — инициализировать состояние объекта, обычно путем установки значений его переменных экземпляра. С другой стороны, метод — это обычная функция, связанная с объектом или классом, которую можно вызывать для выполнения какой-либо конкретной задачи.
58.	17.Что такое статический метод? Чем он отличается от метода экземпляра? Ответ. Статический метод — это метод, который принадлежит самому классу, а не какому-либо экземпляру класса. Это означает, что его можно вызывать непосредственно в классе, без необходимости сначала создавать экземпляр класса. С другой стороны,

	метод экземпляра связан с экземпляром класса и может быть вызван только для объекта класса.
59.	18. В чем разница между частной и защищенной переменной-членом? Ответ. Доступ к закрытой переменной-члену возможен только внутри класса, в котором она определена. Доступ к защищенной переменной-члену можно получить в классе, который ее определяет, а также в любых производных от него подклассах.
60.	20. Что такое переопределение метода? Чем это отличается от перегрузки методов? Ответ: Переопределение метода — это процесс, посредством которого подкласс предоставляет другую реализацию метода, который уже определен в его суперклассе. Это позволяет подклассу изменять поведение метода без изменения его сигнатуры. С другой стороны, перегрузка метода относится к процессу определения нескольких методов с одинаковым именем, но разными параметрами.

3.2.2 Шифр и наименование компетенции

ПКв-3 Способность интеграции программных модулей и компонент и верификации выпуска программных продуктов

Номер вопроса	Текст вопроса
61.	21. Что такое инкапсуляция в ООП? Ответ: Инкапсуляция — это процесс сокрытия внутренних деталей объекта от внешнего мира и раскрытия только того, что необходимо другим объектам для взаимодействия с ним. Обычно это достигается за счет использования модификаторов доступа (таких как <code>public</code> , <code>private</code> и <code>protected</code>) для управления видимостью атрибутов и методов объекта.
62.	22. Что такое конструктор в Java? Ответ: Конструктор в Java — это специальный метод, который вызывается при создании объекта класса. Его цель — инициализировать состояние объекта, как правило, путем установки значений его переменных экземпляра.
63.	23. Что такое класс в Java? Ответ: Класс в Java — это план или шаблон для создания объектов. Он определяет свойства и методы, которыми будут обладать объекты, и предоставляет структуру для организации кода.
64.	24. Что такое объект в Java? Ответ: Объект в Java — это экземпляр класса. Он представляет собой конкретный экземпляр класса с собственным набором значений атрибутов класса.
65.	25. В чем разница между классом и объектом в Java? Ответ: Класс в Java — это план или шаблон для создания объектов, а объект — это экземпляр класса. Другими словами, класс определяет свойства и методы, которые будут иметь объекты этого класса, в то время как объект представляет конкретный экземпляр класса.
66.	26. Что такое статическая переменная в Java? Ответ. Статическая переменная в Java — это переменная, связанная с самим классом, а не с каким-либо конкретным экземпляром класса. Это означает, что все экземпляры класса имеют одно и то же значение статической переменной.
67.	27. Что такое статический метод в Java? Ответ. Статический метод в Java — это метод, принадлежащий самому классу, а не какому-либо экземпляру класса. Это означает, что его можно вызывать непосредственно в классе, без необходимости сначала создавать экземпляр класса.
68.	28. В чем разница между абстрактным классом и интерфейсом в Java? Ответ. Абстрактный класс в Java — это класс, который не может быть создан и предназначен для создания подклассов. Он может содержать один или несколько абстрактных методов, которые объявлены, но не реализованы. Интерфейс, с другой

	<p>стороны, представляет собой набор сигнатур методов, которые могут быть реализованы любым классом, реализующим интерфейс. В отличие от абстрактного класса интерфейс не может содержать никаких деталей реализации, и все методы должны быть реализованы реализующим классом.</p>
69.	<p>29. Что такое полиморфизм в Java? Ответ: Полиморфизм в Java — это способность объекта принимать различные формы. В Java это обычно достигается за счет наследования и интерфейсов.</p>
70.	<p>30. Что такое наследование в Java? Ответ: Наследование в Java — это механизм, с помощью которого один класс (подкласс) может наследовать свойства и методы другого класса (надкласса). Это позволяет подклассу повторно использовать код суперкласса, а также добавлять или изменять собственные функции.</p>
71.	<p>31. Что такое перегрузка методов в Java? Ответ. Перегрузка методов в Java — это процесс определения нескольких методов с одинаковыми именами, но разными параметрами. Это позволяет использовать одно имя метода для разных операций, что делает код более читабельным и простым в обслуживании.</p>
72.	<p>32. Что такое переопределение метода в Java? Ответ: Переопределение метода в Java — это процесс, посредством которого подкласс предоставляет другую реализацию метода, который уже определен в его суперклассе. Это позволяет подклассу изменять поведение метода без изменения его сигнатуры.</p>
73.	<p>33. В чем разница между перегрузкой метода и переопределением метода в Java? Ответ. Перегрузка методов в Java включает определение нескольких методов с одинаковыми именами, но разными параметрами, в то время как переопределение методов предполагает предоставление разных методов.</p>
74.	<p>34. Какова цель ключевого слова final в Java? Ответ. Ключевое слово final в Java может использоваться для того, чтобы сделать переменную, метод или класс константой, что означает, что их значение или поведение нельзя изменить. Его также можно использовать для предотвращения создания подкласса класса или переопределения метода.</p>
75.	<p>35. Что такое пакет в Java? Ответ: Пакет в Java — это способ организации связанных классов и интерфейсов. Это позволяет группировать классы на основе их функциональности и помогает избежать конфликтов имен между классами с похожими именами.</p>
76.	<p>36. Что такое ключевое слово super в Java? Ответ: Ключевое слово super в Java используется для обозначения суперкласса текущего объекта. Его можно использовать для вызова конструктора суперкласса, доступа к его переменным и методам экземпляра или для вызова метода, который был переопределен в подклассе.</p>
77.	<p>37. Что такое это ключевое слово в Java? Ответ: Ключевое слово this в Java используется для ссылки на текущий объект. Его можно использовать для доступа к переменным и методам экземпляра объекта или для вызова его конструктора.</p>
78.	<p>38. Какова цель оператора instanceof в Java? Ответ. Оператор instanceof в Java используется для определения того, является ли объект экземпляром определенного класса или интерфейса. Он возвращает true, если объект является экземпляром указанного класса или его подкласса, или если он реализует указанный интерфейс.</p>
79.	<p>39. Что такое одноэлементный шаблон в Java? Ответ: шаблон singleton в Java — это шаблон проектирования, который гарантирует, что класс имеет только один экземпляр, и предоставляет глобальную точку доступа к этому</p>

	экземпляру. Это может быть полезно в ситуациях, когда должен быть только один экземпляр определенного объекта, такого как соединение с базой данных или диспетчер конфигурации.
80.	40. Что такое фабричный шаблон в Java? Ответ. Фабричный шаблон в Java — это шаблон проектирования, предоставляющий интерфейс для создания объектов, но позволяющий подклассам решать, экземпляр какого класса создавать. Это может быть полезно в ситуациях, когда клиентский код не должен касаться деталей создания объекта и когда соответствующий класс для создания экземпляра может зависеть от условий выполнения

3.3 Кейс-задание

3.3.1 Шифр и наименование компетенции

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

Номер вопроса	Текст вопроса
81.	Класс Дробное число со знаком (Fractions). Число должно быть представлено двумя полями: целая часть - длинное целое со знаком, дробная часть - беззнаковое короткое целое. Реализовать арифметические операции сложения, вычитания, умножения и операции сравнения. В функции main проверить эти методы.
82.	Пирамидальная сортировка Реализовать классический алгоритм пирамидальной сортировки с набором тестов. вход: {5,4,3,2,1} выход: {1,2,3,4,5}
83.	Дерево (tree) Реализовать класс контейнер «Тее»(дерево) с поддержкой следующих операций: 1) итератор для перемещения по дереву; 2) добавить/удалить элемент; 3) получить/добавить поддереву; 4) реализовать итераторы совместимые со стандартными реализующие обход в глубину и ширину Реализовать поддержку ConcurrentModificationException. вход: setRoot «НГУ» add «НГУ» «ММФ» add «НГУ» «ФФ» add«НГУ» «ФИТ» add «НГУ» «ГГФ» add «ФИТ» «СИ» add «ФИТ» «КОИ» выход:
84.	Стек (stack) Реализовать класс стек с поддержкой операций push (добавить элемент), pop(взять элемент), и count(узнать количество элементов). Элементы стека могут быть разных типов. Должна быть возможность использовать стандартный итератор для обхода контейнера. вход: push(2) push(7) pop count выход: {2}
85.	Класс Деньги для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа unsigned char - для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения. В функции main проверить эти методы.
86.	Для выполнения заданий необходимо создать аккаунт на публичном интернет ресурсе github.com и репозиторий «ООР». Репозиторий должен быть доступен для чтения лектору(alexander-a-vlasov) и преподавателю ведущему семинарские занятия (alex-ks). Задания выполняются на языке Java в программном средстве IntelliJ IDEA версии Community. Все решения должны сопровождаться набором тестов проверяющего корректность выполненного задания. Для выполнения первого задания «Пирамидальная сортировка» необходимо создать проект Task_1_1. Следующие задания именовать соответствующим образом. Зеленым цветом обозначены примеры к заданиям, которые можно использовать в качестве первого теста. Серым цветом обозначены дополнительные требования к задаче, которые не обязательны к реализации, но приносят дополнительные баллы Желтым цветом отмечены комментарии к задаче. <u>Дополнительные условия и ограничения.</u>
87.	Класс Равнобокая трапеция, члены класса: координаты 4-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура равнобокой трапецией; вычисления и вывода сведений о фигуре: длины сторон, периметр, площадь. В функции main продемонстрировать работу с классом: дано N трапеций, найти количество трапеций, у которых площадь больше средней площади.
88.	Описать базовый класс Строка. Обязательные поля класса: указатель рChar хранит адрес динамически выделенной памяти для размещения символов строки; значение типа word хранит длину строки в байтах. Реализовать обязательные методы следующего назначения: -конструктор без параметров; -конструктор, принимающий в качестве параметра строковый литерал; п конструктор,

	<p>принимающий в качестве параметра символ; -метод получения длины строки; -метод очистки строки (делает строку пустой); -деструктор. Описать производный от Строка класс Комплексное число. Строки данного класса должны состоять из двух полей разделочных символом i. Первое поле задает значение действительной части числа, а второе - значение мнимой. Каждое из полей может содержать только символы десятичных цифр и символы - и +, определяющие знак числа. Символы - или + могут находиться только в первой позиции числа, причем символ + может отсутствовать, в этом случае число считается положительным. Если в составе инициализирующей строки будет встречен любой символ, отличный от допустимых, объект класса Комплексное число должен принимать нулевое значение. Примеры строк: 33il2, -7U00, +5i-21. Для класса Комплексное_число определить следующие методы: -проверка на равенство; -сложение чисел; -умножение чисел. Написать тестовую программу, которая демонстрирует работу методов базового и производного классов.</p>
89.	<p>Описать шаблонный класс List для работы с однонаправленными списками в динамической памяти. Для объектов класса List определить операции проверки списка на пустоту, добавления элемента в начало списка, в конец списка, подсчет числа вхождений элемента в список, удаление элемента из списка. Продемонстрировать работу с шаблонным классом для списка с целыми элементами и с элементами-строками.</p>

3.3.2 Шифр и наименование компетенции

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

Номер вопроса	Текст вопроса
90.	<p>Григорианский календарь Реализовать класс григорианского календаря: 1) високосный год (28/29 февраля) 2) завершающий год века, если он не делится на 400, не является високосным 3) день недели Выполнить операции с календарем: 1) Какой день недели будет через 1024 дня? 2) Сколько лет, месяцев и дней назад был день победы 9 мая 1945 года? 3) В какой день недели вы родились? 4) Какой месяц будет через 17 недель? 5) Сколько дней до нового года? 6) Ближайшая пятница 13-го числа месяца? Использование классов java.util.Date, java.util.Calendar из пакета java.time запрещено</p>
91.	<p>Зачетная книжка Реализовать класс электронная зачетной книжки студента ФИТ и обеспечить следующие функции: 1) текущий средний бал за все время обучения; 2) может ли студент получить «красный» диплом с отличием; 3) будет ли повышенная стипендия в этом семестре. Для первого теста используйте данные из Вашей зачётной книжки Требования для диплома с отличием: • 75% оценок в приложении к диплому(последняя оценка) – «отлично» • Нет в зачетной книжке итоговых оценок «удовлетворительно» • Квалификационная работа защищена на «отлично»</p>
92.	<p>Упорядоченные множества Реализовать библиотеку для выполнения классических операций над объектами: множество с , частично-упорядоченное множество(ЧУМ), решёточно-упорядоченное множество(РУМ), линейно-упорядоченное множество(ЛУМ): 1) проверка выполнения транзитивность для операции сравнения между элементами массива; 2) топологическая сортировка массива по возрастанию (текущий элемент отсортированного массива больше или не сравним чем все предыдущие); 3) поиск максимальных элементов в массиве. С помощью созданной библиотеки решите несколько прикладных задач: 1) Найти самых умных студентов в группе. 2) Зная функцию вероятности победы каждой команды любого футбольного матча на турнире. Составить турнирную таблицу по олимпийской системе так, чтобы заданная команда выиграла как можно больше матчей</p>
93.	<p>Калькулятор Реализовать калькулятор инженера для вещественных чисел. Пользователь вводит на стандартный поток ввода выражение в префиксном виде. Калькулятор вычисляет значение и выводит на стандартный поток вывода. Кроме стандартных</p>

	операций (+, -, *, /) есть набор функций (log, pow, sqrt, sin, cos). С помощью динамической загрузки библиотеки(plugin) переопределить поведение базовых операций. Добавить поддержку работы с градусами и комплексными числами. вход: sin + - 1 2 1 выход: 0
94.	Записная книжка Сделать записную книжку с набором функций доступных с командной строки: • Добавить запись • Удалить запись • Вывести все записи, отсортированные по времени добавления • Вывести отсортированные по времени добавления записи из заданного интервала времени, содержащие в заголовке ключевые слова. Данные записной книги сериализуются в файл формата JSON. Для работы с форматом JSON рекомендуется использовать библиотеки Jackson или Gson. Для анализа параметров командной строки, так же рекомендуется использовать сторонние библиотеки. Примеры: notebook -add "Моя заметка" "Очень важная заметка" notebook -rm "Моя заметка" notebook -show notebook -show "14.12.2019 7:00" "17.12.2019 13:00" "Моя" "Твоя" "мне"
95.	Биржа Реализовать модель биржи. Биржа — место где «Брокеры» могут совершать операции по продаже и покупке «Товара». Цель брокера за счет проведения операций увеличить свое состояние. Каждый брокер в своем распоряжении имеет определенное количество денег и товара. Операция выполняется за счет того, что вы размещаете свои предложения по покупке и продаже товара на бирже и анализируя текущие предложения можете совершить сделку. В качестве брокера может выступать как человек управляющий командами с консоли, так и несколько типов искусственных интеллектов, реализующих свое поведение согласно стратегии: «Большой куш» — покупают и продают только если есть предложение превышающее стоимость затраченных денег на покупку заданный порог или выполняют минимизацию убытков если в течении длительного времени заработать не получилось. «Игрок» — покупает и продает как можно чаще. «Аналитик» — анализирует историю предложений на бирже и исходя из этой информации совершает сделки. За каждую минуту размещенного на бирже заказа, биржа забирает с брокера фиксированную сумму денег за свои услуги. В системе каждый брокер должен быть представлен независимой нитью исполнения в которой могут асинхронно приниматься решения.
96.	Простое число У вас имеется массив целых чисел, необходимо определить есть ли в этом массиве хотя бы одно не простое (делится без остатка только на себя и единицу). Необходимо предоставить две реализации с последовательным и параллельным решением. Удалось ли ускорить решение задачи за счет применения параллельных вычислений? вход: {6,8,7,13,9,4} выход: True вход: 6997901 6997927 6997937 6997967 6998009 6998029 6998039 6998051 6998053 выход: False Реализовать с помощью ThreadPool без использования StreamAPI

3.4 Подготовка к практической работы

3.4.1 Шифр и наименование компетенции

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

Номер вопроса	Текст вопроса
97.	Введение в объектно-ориентированные методологии (технологии) разработки программных систем. Инструментальные средства, поддерживающие эти технологии ООП
98.	Объектно-ориентированный анализ и проектирование: основные понятия и терминология. Цели анализа и проектирования. Сопоставление ОО языков программирования C# и Java
99.	Основные принципы инкапсуляции. Реализация инкапсуляции в ОО языках программирования C# и Java
100.	Основные принципы и правила наследования. Реализация наследования в ОО языках программирования C# и Java
101.	Основные принципы и правила полиморфизма. Реализация наследования в ОО языках программирования C# и Java
102.	Основы UML. Основные типы связей языка UML. Диаграммы вариантов использования. Язык UML. Основные элементы языка UML.

Критерии и шкалы оценки:

- **оценка «зачтено»** выставляется студенту, если домашнее задание является самостоятельным, оригинальным текстом, в котором прослеживается авторская позиция, продуманная система аргументов, а также наличествует обоснованные выводы; используются термины, понятия по дисциплине, в рамках которой выполняется работа; полностью соответствует выбранной теме, цели и задачам; текст домашнего задания логически выстроен, имеет четкую структуру; работа соответствует всем техническим требованиям; домашнее задание выполнено в установленный срок.

- оценка «не зачтено», выставляется студенту, если домашнее задание не является самостоятельным, оригинальным текстом, в котором не прослеживается авторская позиция, не продумана система аргументов, а также отсутствуют обоснованные выводы; не используются термины, понятия по дисциплине, в рамках которой выполняется работа; не соответствует выбранной теме, цели и задачам; текст домашнего задания композиционно не выстроен; работа не соответствует техническим требованиям; домашнее задание не выполнено в установленный срок.

3.4.1 Шифр и наименование компетенции

ПКв-3 Способность интеграции программных модулей и компонент и верификации выпуска программных продуктов

Номер вопроса	Текст вопроса
103.	Понятие объектно-ориентированного анализа. Диаграммы классов. Диаграммы взаимодействия. Диаграммы состояний.
104.	Основные принципы объектно-ориентированного подхода к созданию пользовательского интерфейса. Реализация в ОО языках программирования C# и Java
105.	Принципы объектно-ориентированного проектирования программ. Реализация в ОО языках программирования C# и Java
106.	Диаграммы пакетов, компонентов и размещения. Разработка моделей реальных и концептуальных систем на основе методологии компонентно-ориентированного программирования в ОО языках программирования C# и Java

3.5 Курсовая работа

3.5.1 Шифр и наименование компетенции

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

Номер вопроса	Текст вопроса
107.	Разработка программного приложения «Покупка автомобиля в автосалоне» в среде объектно-ориентированного программирования «Delphi»
108.	Реализация контейнера на основе комбинированной структуры "Стек-массив динамических списков"
109.	Разработка объектной программы для задачи "Строительная организация"
110.	Объектная реализация контейнера на основе комбинированной структуры «Двухнаправленный список массивов-стеков» - Книжный магазин (Delphi, Pascal)
111.	Объектная реализация контейнера на основе комбинированной структуры "Упорядоченный массив динамических очередей"
112.	Двухнаправленный список
113.	Объектная реализация контейнера на основе комбинированной структуры «Динамический список массивов-стеков
114.	Массив динамических очередей
115.	Система моделирования электрических цепей
116.	. Система разработки схемы компьютерной сети
117.	Редактор трехмерных сцен с использованием <i>Direct3D</i>
118.	Элементы графического пользовательского интерфейса с использованием библиотеки <i>OpenGL</i>

3.5.1 Шифр и наименование компетенции

ПКв-3 Способность интеграции программных модулей и компонент и верификации выпуска программных продуктов

Номер вопроса	Текст вопроса
119.	. Однотабличная СУБД с обработкой основных SQL запросов
120.	Редактор трехмерных сцен с использованием <i>OpenGL</i>
121.	Редактор векторной графики
122.	Библиотека графического пользовательского интерфейса с использованием <i>DirectX</i>
123.	Клиент базы данных с использованием библиотеки <i>MFC (ODBC)</i>
124.	Клиент базы данных с использованием библиотеки <i>MFC (ADO/DAO)</i>
125.	Редактор трехмерных сцен с использованием <i>Direct3D</i>
126.	Электронное обучающее пособие по операторам языка C++

127.	Универсальный построитель графиков функций
128.	Электронное обучающее пособие по инструкциям языка C++
129.	Электронное обучающее пособие по контейнерным классам STL
130.	Электронное обучающее пособие по библиотеке ввода-вывода C++
131.	Объектно-ориентированная библиотека основных алгоритмов линейной алгебры
132.	Поисковая система на основе хэш-таблиц
133.	Редактор блок-схем
134.	Информационная система склада предприятия

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

Процедуры оценивания в ходе изучения дисциплины знаний, умений и навыков, характеризующих этапы формирования компетенций, регламентируются положениями:

- П ВГУИТ 2.4.03 Положение о курсовых экзаменах и экзаменах;
- П ВГУИТ 4.1.02 Положение о рейтинговой оценке текущей успеваемости.

Для оценки знаний, умений, навыков обучающихся по дисциплине применяется рейтинговая система. Итоговая оценка по дисциплине определяется на основании определения среднеарифметического значения баллов по каждому заданию.

Зачет по дисциплине выставляется в зачетную ведомость по результатам работы в семестре после выполнения всех видов учебной работы, предусмотренных рабочей программой дисциплины (с отметкой «зачтено») и получении по результатам тестирования по всем разделам дисциплины не менее 60 %.

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

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					
Знает	основные теоретико-методологические положения системного подхода любой категории информации принципы сбора, отбора и обобщения информации	Результаты тестирования	Обучающимся даны правильные ответы менее чем на 59,99 % всех тестовых вопросов	Неудовлетворительно	Не освоена / недостаточный
			Обучающимся даны правильные ответы на 60-74,99% всех тестовых вопросов	Удовлетворительно	Освоена / базовый
			Обучающимся даны правильные ответы на 75-84,99% всех тестовых вопросов	Хорошо	Освоена / повышенный
			Обучающимся даны правильные ответы на 85-100% всех тестовых вопросов	Отлично	Освоена / повышенный
		Собеседование (зачет / экзамен)	Обучающийся обладает частичными и разрозненными знаниями, только некоторые из которых может связывать между собой	Неудовлетворительно	Не освоена / недостаточный
			Обучающийся обладает минимальным набором знаний, необходимым для системного взгляда на изучаемый объект	Удовлетворительно	Освоена / базовый
			Обучающийся обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Хорошо	Освоена / повышенный
			Обучающийся обладает системным взглядом на изучаемый объект	Отлично	Освоена / повышенный
Умеет	осуществлять поиск информации для решения поставленной задачи по различным типам запросов и сопоставлять разные источники информации с целью выявления их противоречий и поиска достоверных суждений соотносить разнородные	Отчет по практическим работам, реферат	Обучающийся не владеет умениями выполнения заданий; не демонстрирует умений, предусмотренных планируемыми результатами обучения	Неудовлетворительно	Не освоена / недостаточный
			Обучающийся испытывает затруднения при выполнении заданий по алгоритму; демонстрирует минимальный набор умений, предусмотренных планируемыми результатами обучения	Удовлетворительно	Освоена / базовый
			Обучающийся выполняет задания с использованием алгоритма решения, при выполнении допускает незначительные ошибки и неточности, формулирует выводы; демонстрирует умения, предусмотренные планируемыми результатами обучения	Хорошо	Освоена / повышенный

	явления и систематизировать их в рамках избранных видов профессиональной деятельности		Обучающийся выполняет задания, формируя алгоритм решения, при выполнении не допускает ошибок и неточностей, формулирует выводы; демонстрирует умения, предусмотренные планируемыми результатами обучения	Отлично	Освоена / повышенный
Владеет	методами анализа информации и предлагает возможные варианты решения поставленной задачи, оценивая их достоинства и недостатки навыками работы с информационными источниками, опыт научного поиска, создания научных текстов	Кейс-задание,	Обучающийся не владеет умениями выполнения заданий; не демонстрирует умений, предусмотренных планируемыми результатами обучения	Неудовлетворительно	Не освоена / недостаточный
			Обучающийся испытывает затруднения при выполнении заданий по алгоритму; демонстрирует минимальный набор умений, предусмотренных планируемыми результатами обучения	Удовлетворительно	Освоена / базовый
			Обучающийся выполняет задания с использованием алгоритма решения, при выполнении допускает незначительные ошибки и неточности, формулирует выводы; демонстрирует умения, предусмотренные планируемыми результатами обучения	Хорошо	Освоена / повышенный
			Обучающийся выполняет задания, формируя алгоритм решения, при выполнении не допускает ошибок и неточностей, формулирует выводы; демонстрирует умения, предусмотренные планируемыми результатами обучения	Отлично	Освоена / повышенный

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 Способность интеграции программных модулей и компонент и верификации выпуска программных продуктов					
Знает	знает регламент проведения разработки процедур интеграции программных модулей и компонент и алгоритмы верификации выпусков программного продукта	Результаты тестирования	Обучающимся даны правильные ответы менее чем на 59,99 % всех тестовых вопросов	Неудовлетворительно	Не освоена / недостаточный
			Обучающимся даны правильные ответы на 60-74,99% всех тестовых вопросов	Удовлетворительно	Освоена / базовый
			Обучающимся даны правильные ответы на 75-84,99% всех тестовых вопросов	Хорошо	Освоена / повышенный
			Обучающимся даны правильные ответы на 85-100% всех тестовых вопросов	Отлично	Освоена / повышенный
		Собеседование (зачет / экзамен)	Обучающийся обладает частичными и разрозненными знаниями, только некоторые из которых может связывать между собой	Неудовлетворительно	Не освоена / недостаточный
			Обучающийся обладает минимальным набором знаний, необходимым для системного взгляда на изучаемый объект	Удовлетворительно	Освоена / базовый
			Обучающийся обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Хорошо	Освоена / повышенный
			Обучающийся обладает системным взглядом на изучаемый объект	Отлично	Освоена / повышенный
Умеет	: проводить разработку процедур интеграции программных модулей; программных модулей и компонент и верификацию выпусков программного продукта	Отчет по практическим работам, реферат	Обучающийся не владеет умениями выполнения заданий; не демонстрирует умений, предусмотренных планируемыми результатами обучения	Неудовлетворительно	Не освоена / недостаточный
			Обучающийся испытывает затруднения при выполнении заданий по алгоритму; демонстрирует минимальный набор умений, предусмотренных планируемыми результатами обучения	Удовлетворительно	Освоена / базовый
			Обучающийся выполняет задания с использованием алгоритма решения, при выполнении допускает незначительные ошибки и неточности, формулирует выводы; демонстрирует умения, предусмотренные планируемыми результатами обучения	Хорошо	Освоена / повышенный
			Обучающийся выполняет задания, формируя алгоритм решения, при выполнении не допускает ошибок и неточностей, формулирует выводы; демонстрирует умения, предусмотренные	Отлично	Освоена / повышенный

			планируемыми результатами обучения		
Владеет	методами и алгоритмами разработки процедур интеграции программных модулей методами и алгоритмами интеграции программных модулей и компонент и верификации выпусков программного продукта	Курсовая работа	Обучающийся не владеет умениями выполнения заданий; не демонстрирует умений, предусмотренных планируемыми результатами обучения	Неудовлетворительно	Не освоена / недостаточный
			Обучающийся испытывает затруднения при выполнении заданий по алгоритму; демонстрирует минимальный набор умений, предусмотренных планируемыми результатами обучения	Удовлетворительно	Освоена / базовый
			Обучающийся выполняет задания с использованием алгоритма решения, при выполнении допускает незначительные ошибки и неточности, формулирует выводы; демонстрирует умения, предусмотренные планируемыми результатами обучения	Хорошо	Освоена / повышенный
			Обучающийся выполняет задания, формируя алгоритм решения, при выполнении не допускает ошибок и неточностей, формулирует выводы; демонстрирует умения, предусмотренные планируемыми результатами обучения	Отлично	Освоена / повышенный