

Подписано цифровой подписью: АНОО ВО
"СИБИТ"

Причина: Я утвердил этот документ
DN: ИИН ЮЛ=7707329152, E=uc@tax.gov.ru,
ОГРН=1047707030513, C=RU, S=77 Москва, L=г.
Москва, STREET="ул. Неглинная, д. 23",
O=Федеральная налоговая служба, CN=Федеральная
налоговая служба

УТВЕРЖДЕНО:

Ректор

Родионов М. Г.

(протокол от 28.08.2024 № 12)

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»**

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.03 Прикладная информатика

Направленность (профиль) подготовки: Прикладная информатика в экономике

Квалификация (степень) выпускника: бакалавр

Формы обучения: очная, очно-заочная, заочная

Год набора (приема на обучение): 2024

Срок получения образования: Очная форма обучения – 4 года
 Очно-заочная форма обучения – 4 года 10 месяца(-ев)
 Заочная форма обучения – 4 года 10 месяца(-ев)

Объем: в зачетных единицах: 7 з.е.
 в академических часах: 252 ак.ч.

г. Омск, 2024

Разработчики:

Старший преподаватель, факультет очного обучения
Куликова Е. В.

**Рецензенты:**

Ультан А.Е., доцент кафедры «Информационная безопасность» Омского государственного университета путей сообщения, к.т.н.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки Направление подготовки: 09.03.03 Прикладная информатика, утвержденного приказом Минобрнауки России от 19.09.2017 №922, с учетом трудовых функций профессиональных стандартов: "Программист", утвержден приказом Минтруда России от 20.07.2022 № 424н; "Специалист по информационным системам", утвержден приказом Минтруда России от 13.07.2023 № 586н; "Руководитель проектов в области информационных технологий", утвержден приказом Минтруда России от 27.04.2023 № 369н; "Системный аналитик", утвержден приказом Минтруда России от 27.04.2023 № 367н.

Согласование и утверждение

№	Подразделение или коллегиальный орган	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1		Руководитель образовательной программы	Родионов М. Г.	Согласовано	28.08.2024, № 12

Содержание (рабочая программа)

1. Цель и задачи освоения дисциплины (модуля)
2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы
3. Место дисциплины в структуре ОП
4. Объем дисциплины и виды учебной работы
5. Содержание дисциплины
 - 5.1. Разделы, темы дисциплины и виды занятий
 - 5.2. Содержание разделов, тем дисциплины
6. Рекомендуемые образовательные технологии
7. Оценочные материалы текущего контроля
8. Оценочные материалы промежуточной аттестации
9. Порядок проведения промежуточной аттестации
10. Материально-техническое и учебно-методическое обеспечение дисциплины
 - 10.1. Перечень основной и дополнительной учебной литературы
 - 10.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся
 - 10.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине
 - 10.4. Специальные помещения, лаборатории и лабораторное оборудование
11. Методические указания по освоению дисциплины (модуля)

1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - овладение студентами комплексом знаний по теоретическим и прикладным основам объектно-ориентированного программирования; формирование умений разработки, внедрения, адаптации программ на объектно-ориентированном языке; развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений

Задачи изучения дисциплины:

- формирование представлений о разработке приложений и систем, основанных на объектно-ориентированном подходе;
- ознакомление с базовыми понятиями объектно-ориентированного программирования;
- ознакомление с инструментальным ПО разработки приложений и систем, основанных на объектно-ориентированном подходе;
- формирование умений разработки, внедрения, тестирования и адаптации программных средств;
- формирование умений интеграции программных модулей и компонент программного обеспечения;
- формирование базы для изучения дисциплин, имеющих межпредметную связь.

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

Компетенции, индикаторы и результаты обучения

УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений

УК-2.1 Знает необходимые для осуществления профессиональной деятельности правовые нормы и методологические основы принятия управленческого решения.

Знать:

УК-2.1/Зн8 Методику определения цели и задач, разработки плана и целевых этапов создания программы

УК-2.1/Зн9 Принципы выбора оптимальных языков и инструментов разработки программ

УК-2.2 Умеет анализировать альтернативные варианты решений для достижения намеченных результатов; разрабатывать план, определять целевые этапы и основные направления работ.

Уметь:

УК-2.2/Ум7 Выбирать оптимальные языки и инструментальные средства разработки программ, исходя из имеющихся программных, аппаратных ресурсов и ограничений

УК-2.2/Ум8 Анализировать альтернативные варианты реализации алгоритмов для достижения намеченных результатов и реализации программных решений

УК-2.2/Ум9 Формулировать цели и задачи проекта по созданию программы

ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности

ОПК-2.1 Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности.

Знать:

ОПК-2.1/Зн6 Технологии и методы объектно-ориентированного программирования

ОПК-2.1/Зн7 Принципы объектно-ориентированного подхода

ОПК-2.1/Зн8 Основные языки объектно-ориентированного программирования

ОПК-2.1/Зн9 Информационные технологии и программные средства объектно-ориентированного подхода

ОПК-2.2 Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности.

Уметь:

ОПК-2.2/Ум7 Осуществлять выбор программного обеспечения для разработки приложений на объектно-ориентированном языке

ОПК-2.2/Ум8 Настраивать программную среду для разработки программ

ОПК-2.2/Ум9 Применять технологии и методы объектно-ориентированного программирования в процессе разработки программного средства

ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения

ОПК-7.1 Знает основные алгоритмические конструкции, языки и системы программирования, современные программные среды разработки программного обеспечения информационных систем и технологий.

Знать:

ОПК-7.1/Зн5 Современные программные среды разработки приложений на объектно-ориентированном языке

ОПК-7.1/Зн6 Лексику, синтаксис, основные конструкции, классы и интерфейсы объектно-ориентированного языка.

ОПК-7.1/Зн7 Практические приемы разработки приложений

ОПК-7.2 Умеет разрабатывать алгоритмы и применять языки программирования и современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных систем.

Уметь:

ОПК-7.2/Ум5 Разрабатывать консольные приложения на объектно-ориентированном языке, выполнять их тестирование и отладку

ОПК-7.2/Ум6 Разрабатывать приложения с GUI-интерфейсом для автоматизации бизнес-процессов, решения прикладных задач различных классов

ОПК-7.2/Ум7 Разрабатывать приложения для работы с базами данных и информационными системами

3. Место дисциплины в структуре ОП

Дисциплина (модуль) «Объектно-ориентированное программирование» относится к обязательной части образовательной программы и изучается в семестре(ах): Очная форма обучения - 5, 6, Очно-заочная форма обучения - 6, 7, Заочная форма обучения - 6, 7.

Предшествующие дисциплины (практики) по связям компетенций:

- Алгоритмизация и программирование;
- Информатика и информационные технологии;
- Ознакомительная практика;
- Операционные системы и среды;
- Правоведение;
- Средства и системы проектирования ПО;
- Учебно-воспитательный семинар;

Последующие дисциплины (практики) по связям компетенций:

- Выполнение и защита выпускной квалификационной работы;
- Информационные технологии в экономике и управлении;
- Средства и системы проектирования ПО;
- Технологическая (преддипломная) практика;
- Технологическая (проектно-технологическая) практика;
- Управление IT-проектами;
- Управленческие решения;
- Учебно-воспитательный семинар;

В процессе изучения дисциплины студент готовится к видам профессиональной деятельности и решению профессиональных задач, предусмотренных ФГОС ВО и образовательной программой.

4. Объем дисциплины и виды учебной работы

Очная форма обучения

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Контактная работа (часы, всего)	Консультации (часы)	Лабораторные занятия (часы)	Лекционные занятия (часы)	Практические занятия (часы)	Самостоятельная работа (часы)	Промежуточная аттестация (часы)
Пятый семестр	144	4	74	2	36	18	18	61	Зачет (9)
Шестой семестр	108	3	58	4	36	18		23	Экзамен (27)
Всего	252	7	132	6	72	36	18	84	36

Очно-заочная форма обучения

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Контактная работа (часы, всего)	Консультации (часы)	Лабораторные занятия (часы)	Лекционные занятия (часы)	Практические занятия (часы)	Самостоятельная работа (часы)	Промежуточная аттестация (часы)

Шестой семестр	144	4	52	4	24	12	12	88	Зачет (4)
Седьмой семестр	108	3	38	2	12	12	12	61	Экзамен (9)
Всего	252	7	90	6	36	24	24	149	13

Заочная форма обучения

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Контактная работа (часы, всего)	Консультации (часы)	Лабораторные занятия (часы)	Лекционные занятия (часы)	Практические занятия (часы)	Самостоятельная работа (часы)	Промежуточная аттестация (часы)
Шестой семестр	144	4	12	4	4	2	2	128	Зачет (4)
Седьмой семестр	108	3	8	2	2	2	2	91	Экзамен (9)
Всего	252	7	20	6	6	4	4	219	13

5. Содержание дисциплины

5.1. Разделы, темы дисциплины и виды занятий (часы промежуточной аттестации не указываются)

Очная форма обучения

Наименование раздела, темы	Всего	Консультации	Лабораторные занятия	Лекционные занятия	Практические занятия	Самостоятельная работа	Планируемые результаты обучения, соответствующие результатам освоения программы
Раздел 1. Часть 1	135	2	36	18	18	61	УК-2.1 УК-2.2 ОПК-2.1 ОПК-2.2 ОПК-7.1 ОПК-7.2
Тема 1.1. Основы объектно-ориентированного программирования. Введение в технологию JAVA	22		8	4	4	6	
Тема 1.2. Лексика языка Java	20		4	2	2	12	
Тема 1.3. Типы данных	28		8	4	4	12	
Тема 1.4. Code conventions for the JAVA (CCJ)	23		4	2	2	15	
Тема 1.5. Классы, библиотеки, интерфейсы, пакеты	42	2	12	6	6	16	
Раздел 2. Часть 2	81	4	36	18		23	УК-2.1 УК-2.2 ОПК-2.1
Тема 2.1. Классы, библиотеки, интерфейсы, пакеты	14		4	2		8	

Тема 2.2. Java API	33		14	10		9	ОПК-2.2 ОПК-7.1 ОПК-7.2
Тема 2.3. Дженерики и коллекции	18		6	6		6	
Тема 2.4. Разработка приложения с GUI-интерфейсом	16	4	12				
Итого	216	6	72	36	18	84	

Очно-заочная форма обучения

Наименование раздела, темы	Всего	Консультации	Лабораторные занятия	Лекционные занятия	Практические занятия	Самостоятельная работа	Планируемые результаты обучения, соответствующие результатам освоения программы
Раздел 1. Часть 1	140	4	24	12	12	88	УК-2.1 УК-2.2 ОПК-2.1 ОПК-2.2 ОПК-7.1 ОПК-7.2
Тема 1.1. Основы объектно-ориентированного программирования. Введение в технологию JAVA	26		4	2	2	18	
Тема 1.2. Лексика языка Java	28		4	2	4	18	
Тема 1.3. Типы данных	26		4	2	2	18	
Тема 1.4. Code conventions for the JAVA (CCJ)	32		8	4	2	18	
Тема 1.5. Классы, библиотеки, интерфейсы, пакеты	28	4	4	2	2	16	
Раздел 2. Часть 2	99	2	12	12	12	61	УК-2.1 УК-2.2 ОПК-2.1 ОПК-2.2 ОПК-7.1 ОПК-7.2
Тема 2.1. Классы, библиотеки, интерфейсы, пакеты	23		2	4	2	15	
Тема 2.2. Java API	23		4	2	2	15	
Тема 2.3. Дженерики и коллекции	27		4	4	4	15	
Тема 2.4. Разработка приложения с GUI-интерфейсом	26	2	2	2	4	16	
Итого	239	6	36	24	24	149	

Заочная форма обучения

Наименование раздела, темы	Всего	Консультации	Лабораторные занятия	Лекционные занятия	Практические занятия	Самостоятельная работа	Планируемые результаты обучения, соответствующие результатам освоения программы
Раздел 1. Часть 1	140	4	4	2	2	128	УК-2.1

Тема 1.1. Основы объектно-ориентированного программирования. Введение в технологию JAVA	24		2			22	УК-2.2 ОПК-2.1 ОПК-2.2 ОПК-7.1 ОПК-7.2
Тема 1.2. Лексика языка Java	26	2				24	
Тема 1.3. Типы данных	26				2	24	
Тема 1.4. Code conventions for the JAVA (CCJ)	30	2				28	
Тема 1.5. Классы, библиотеки, интерфейсы, пакеты	34		2	2		30	
Раздел 2. Часть 2	99	2	2	2	2	91	УК-2.1 УК-2.2 ОПК-2.1 ОПК-2.2 ОПК-7.1 ОПК-7.2
Тема 2.1. Классы, библиотеки, интерфейсы, пакеты	22			2		20	
Тема 2.2. Java API	27				2	25	
Тема 2.3. Дженерики и коллекции	16	2				14	
Тема 2.4. Разработка приложения с GUI-интерфейсом	34		2			32	
Итого	239	6	6	4	4	219	

5.2. Содержание разделов, тем дисциплин

Раздел 1. Часть 1

Тема 1.1. Основы объектно-ориентированного программирования. Введение в технологию JAVA

1) Введение в объектно-ориентированное программирование. Основные принципы объектно-ориентированного подхода и программирования. Класс и объект. Технологии и методы объектно-ориентированного программирования. Языки объектно-ориентированного программирования. Java. Python. C++. Ruby. C#.

2) Введение в технологию JAVA. История и эволюция Java. Характерные особенности Java. Виртуальная машина и байткод. Интерпретируемость, независимость от платформы, переносимость языка Java. Производительность языка Java. Сборка мусора. Безопасность. Многопоточное и распределенное программирование. Экосистема Java.

Тема 1.2. Лексика языка Java

Лексика языка Java: кодировка, анализ программ, виды лексем (ключевые слова, идентификаторы, литералы, разделители. операторы). Виды операторов. Способы записей. Способы и средства преобразования файлов. Инструменты анализа программ.

Тема 1.3. Типы данных

1) Java – типизированный язык. Простые типы данных. Ссылочные (или объектные) типы. Классификация простых типов. Целочисленные: byte, short, int, long, char. Дробные: float, double. Булевые: boolean. Допустимые операции над аргументами различных типов. Преобразование типов данных в Java.

2) Понятие и виды ссылочных типов. Объект (object). Типы классов. Типы интерфейсов. Типы массивов. Классы, классы-оболочки. Ссылочный строковый тип. Определение классов в программе. Массивы в Java. Массивы элементов базовых типов. Методы класса Arrays. Многомерные массивы.

Тема 1.4. Code conventions for the JAVA (CCJ)

Конвенции кодирования. Code Conventions for the Java (CCJ). CCJ: структура, разделы. Простые предложения (простые операторы). Сложные предложения (составные операторы). Оператор return. Операторы if, if-else, if-else-if-else. Оператор for. Оператор while. Оператор do- while. Отступы, комментарии. Комментарии кода. Комментарии для документирования. Список дескрипторов Javadoc CCJ: декларации, предложения языка, операторы. Пустые строки. Пробелы. Конвенции именования. Практика программирования, приемы программирования.

Тема 1.5. Классы, библиотеки, интерфейсы, пакеты

1) Классы: виды, элементы, объявление, инициализация. Суперклассы. Объявление классов, полей, методов. Спецификация суперкласса. Перегруженные методы. Конструкции классов и области видимости. Спецификаторы доступа (public, protected, private). Разновидности переменных. Конструкторы. Формат объявления конструкторов.

2) Инициализация, инициализаторы. Методы инициализации полей. Инициализация в месте объявления поля. Инициализационный блок.

Раздел 2. Часть 2

Тема 2.1. Классы, библиотеки, интерфейсы, пакеты

Приведение типов классов. Абстрактные классы и абстрактные методы. Интерфейсы. Интерфейсы на основе спецификаций классов. Защищенный доступ к классам. Пакеты. Модификаторы доступа. Пакеты String, FileInputStream, ArrayList, IOException

Тема 2.2. Java API

1) Java Application Programming Interface. Ядро Java API. Типы Java-приложений. Сравнение автономных приложений и апплетов.

2) Графические интерфейсы пользователя (GUI – для standalone applications или applets). Библиотеки GUI. Базовая архитектура AWT. Пакет AWT: иерархия классов. Класс Graphics: основные методы рисования, работа с цветом, шрифтом.

3) Container – абстрактный подкласс класса Component. Класс Panel. Контейнеры ScrollPane, Window, Frame, Dialog. Основные события.

4) Библиотека Swing. JComponent. Классы пакета javax.swing. Окно JFrame. Менеджеры размещения. Менеджер последовательного размещения FlowLayout. Менеджер граничного размещения BorderLayout. Менеджер табличного размещения GridLayout. Менеджер блочного размещения BoxLayout и класс Box.

5) События. Обработка событий. Семантические события и их источники-компоненты. Модель делегирования событий. Низкоуровневые события

Тема 2.3. Дженирики и коллекции

- 1) Дженирики и введение во фреймворк «коллекции».
- 2) Типы коллекций. Интерфейс Collection. Иерархия коллекций. Особенности использования коллекций.
- 3) Принципы объектно-ориентированного дизайна (ООД) классов.

Тема 2.4. Разработка приложения с GUI-интерфейсом

Разработка приложения с GUI-интерфейсом

6. Рекомендуемые образовательные технологии

При преподавании дисциплины применяются разнообразные образовательные технологии в зависимости от вида и целей учебных занятий.

Теоретический материал излагается на лекционных занятиях в следующих формах:

- проблемные лекции;
- лекция-беседа;
- лекции с разбором практических ситуаций.

Семинарские занятия по дисциплине ориентированы на закрепление теоретического материала, изложенного на лекционных занятиях, а также на приобретение дополнительных знаний, умений и практических навыков осуществления профессиональной деятельности посредством активизации и усиления самостоятельной деятельности обучающихся.

Большинство практических занятий проводятся с применением активных форм обучения, к которым относятся:

- 1) устный опрос студентов с элементами беседы и дискуссии по вопросам, выносимым на практические занятия;
- 2) групповая работа студентов, предполагающая совместное обсуждение какой-либо проблемы (вопроса) и выработку единого мнения (позиции) по ней (метод группового обсуждения, круглый стол);
- 3) контрольная работа по отдельным вопросам, целью которой является проверка знаний студентов и уровень подготовленности для усвоения нового материала по дисциплине.

На семинарских занятиях оцениваются и учитываются все виды активности студентов: устные ответы, дополнения к ответам других студентов, участие в дискуссиях, работа в группах, инициативный обзор проблемного вопроса, письменная работа.

7. Порядок проведения промежуточной аттестации

Промежуточная аттестация: Очная форма обучения, Зачет, Пятый семестр.

1. Работа с тестовыми заданиями
2. Выполнение итоговой работы

Промежуточная аттестация: Очная форма обучения, Экзамен, Шестой семестр.

1. Работа с тестовыми заданиями
2. Выполнение итоговой работы

Промежуточная аттестация: Очно-заочная форма обучения, Зачет, Шестой семестр.

1. Работа с тестовыми заданиями
2. Выполнение итоговой работы

Промежуточная аттестация: Очно-заочная форма обучения, Экзамен, Седьмой семестр.

1. Работа с тестовыми заданиями

2. Выполнение итоговой работы

Промежуточная аттестация: Заочная форма обучения, Зачет, Шестой семестр.

1. Работа с тестовыми заданиями
2. Выполнение итоговой работы

Промежуточная аттестация: Заочная форма обучения, Экзамен, Седьмой семестр.

1. Работа с тестовыми заданиями
2. Выполнение итоговой работы

8. Оценочные материалы текущего контроля

Раздел 1. Часть 1

Контролируемые ИДК: УК-2.1 УК-2.2 ОПК-2.1 ОПК-7.1 ОПК-2.2 ОПК-7.2

Тема 1.1. Основы объектно-ориентированного программирования. Введение в технологию JAVA

Форма контроля/оценочное средство: Посещение и работа на лекционных и практических занятиях

Вопросы/Задания:

1. Посещение занятий:
 - а) посещение лекционных и практических занятий,
 - б) соблюдение дисциплины.
2. Работа на лекционных занятиях:
 - а) ведение конспекта лекций,
 - б) уровень освоения теоретического материала,
 - в) активность на лекции, умение формулировать вопросы лектору.
3. Работа на практических занятиях:

Лабораторное занятие 1.

Знакомство с реализацией языков ООП на примере конкретных приложений. Анализ основных характерных особенностей языков. Составление таблицы "Сравнительная характеристика языков ООП". Анализ кодов программ, созданных на языках ООП.

Практическое занятие 1.

Информационные технологии и программные средства объектно-ориентированного подхода. CASE-средства проектирования. Онлайн-инструменты построения диаграмм классов. Редакторы кода. Интегрированные среды разработки. Критерии выбора программного обеспечения для разработки приложений на объектно-ориентированном языке. Выбор языков и инструментальных средств средств, исходя из имеющихся программных, аппаратных ресурсов и ограничений.

Лабораторное занятие 2.

Программная среда: установка инструментария разработчика. Установка интегрированной среды разработки. Настройка программной среды для разработки программ на языке ООП. Первичное знакомство с интерфейсом и функциональными возможностями интегрированной среды разработки.

Лабораторное занятие 3.

Изучение редакций и реализаций Java. Изучение программной среды Java, стандартного инструментария. Работа в командной строке.

Практическое занятие 2.

Контрольная работа №1 "Основы объектно-ориентированного программирования. Введение в

технологииу JAVA"

Лабораторное занятие 4.

Разбор готовых примеров - консольных приложений. Создание первых программ. Пример. Создать программу, выводящую строку приветствия на экран (имя файла с листингом программы: HelloUser.java).

Тема 1.2. Лексика языка Java

Форма контроля/оценочное средство: Посещение и работа на лекционных и практических занятиях

Вопросы/Задания:

1. Посещение занятий:

- а) посещение лекционных и практических занятий,
- б) соблюдение дисциплины.

2. Работа на лекционных занятиях:

- а) ведение конспекта лекций,
- б) уровень освоения теоретического материала,
- в) активность на лекции, умение формулировать вопросы лектору.

3. Работа на практических занятиях:

Лабораторное занятие 1.

Анализ кодов программ на наличие и определение типов лексем. Изучение требований к именованию идентификаторов. Переменные. Объявления переменных примитивного типа. Тренировочные задания: 1) в списке перечислены допустимые и недопустимые идентификаторы переменных. Разбить их на две группы: допустимые/недопустимые; 2) заполнить таблицу ключевых слов.

Лабораторное занятие 2.

Изучение структуры программы. Создание программ, используя различные виды операторов. Преобразование файлов различных кодировок. Анализ программ. Знакомство с утилитой SymbolTest.

Практическое занятие 1.

Контрольная работа №2 "Лексика языка Java"

Тема 1.3. Типы данных

Форма контроля/оценочное средство: Посещение и работа на лекционных и практических занятиях

Вопросы/Задания:

1. Посещение занятий:

- а) посещение лекционных и практических занятий,
- б) соблюдение дисциплины.

2. Работа на лекционных занятиях:

- а) ведение конспекта лекций,
- б) уровень освоения теоретического материала,
- в) активность на лекции, умение формулировать вопросы лектору.

3. Работа на практических занятиях:

Лабораторное занятие 1.

Создание программ с переменными простых типов данных. Выполнение операций над аргументами простых типов.

Пример. Объявите переменные a, b, maxI, minI. Переменной a присвоить строковое значение «123». Переменным maxI, minI присвоить максимальное и минимальное значения, используя класс-обертку Integer. Переменной b присвоить значение, получаемое преобразованием из текста в число. Вывести значения на экран.

Практическое занятие 1.

Преобразование типов: методы, явное преобразование, защита от исключений. Операции явного приведения типов. Сужение. Контрольные вопросы: 1. Перечислите преобразования, которые являются расширяющими. 2. Перечислите преобразования, которые являются сужающими. 3. Что происходит с целочисленным числом при сужении к менее емкому целочисленному? 4. Какие преобразования считаются безопасными? 5. Что происходит с целочисленным числом при расширении к более емкому целочисленному или дробному?

Лабораторное занятие 2.

Создание программ с преобразованием типов.

Пример 1. Объявите и инициализируйте переменные различных типов. Выполните преобразование типов переменных, используя операции явного приведения типов.

Пример 2. Объявите переменные `boolX`, `boolY`, `b1`, `b2`, `b3` (типа `boolean`), `intX`, `intY` (типа `int`). Значениями `boolX`, `boolY` являются результаты вычисления логических выражений. При этом, составьте логические выражения, используя булевы переменные `b1`, `b2`, `b3` и булевы операторы, таким образом, что значением `boolX` является истина, а `boolY` – ложь. Значения переменных `intX`, `intY` определить, используя тернарный оператор. Значениями переменных `intX`, `intY` соответственно являются единицы, в случае, если значения `boolX`, `boolY` – это истина и нули в противном случае.

Лабораторное занятие 3.

Создание программ с определением классов.

Пример 1. Создать программу, рассчитывающую оплату труда рабочего. Оплата труда рассчитывается в зависимости от количества отработанных дней, количества часов в рабочем дне и стоимости часа. Эти данные должны вводиться с клавиатуры. В программе `Program3327.java` должен создаваться экземпляр класса `Work` и выполняться вызов необходимых методов. Все методы должны быть определены в файле `Work.java`. Переменные (количество отработанных дней, количество часов в рабочем дне и стоимость часа) должны быть объявлены в `Work.java`.

Пример 2. Определите класс `Dog` (собака). Атрибутами собаки задайте следующие: кличка, порода, возраст. Атрибуты – это переменные, которые объявляются внутри класса.

Добавьте в класс `Dog` следующие методы:

- 1) метод `eat()`: не возвращает никаких данных – `void`
- 2) метод `walk()`: не возвращает никаких данных – `void`
- 3) метод `sleep()`: не возвращает никаких данных – `void`

Тело данных методов содержит просто вывод сообщения о том, что собака ест/гуляет/спит.

Добавьте в класс `Dog` метод `getAboutDog()`, возвращающий информацию о собаке: кличку, породу, возраст. Этот метод возвращает значение типа `String`.

В программе `Program3322.java` выполните обращение к классу и следующие действия:

- 1) Создайте три объекта на основе класса `Dog`;
- 2) Для каждого объекта присвойте значения переменных;
- 3) Для каждого объекта выполните вывод сообщения с информацией о собаке на основе вызова метода `getAboutDog()`;
- 4) Для каждого объекта вызовите один из методов `eat()`, `walk()`, `sleep()`.

Лабораторное занятие 4.

Создание и обработка массивов.

Пример 1. Создайте массив, содержащий `n` дробных чисел. Выведите элементы массива на консоль в одну строку, разделяя запятой. Размерность массива `n` должна вводиться с клавиатуры. Найдите среднее арифметическое всех элементов массива.

Пример 2. Создайте массив `arr1`, содержащий `n` чисел типа `int`. Размерность массива `n` и значения элементов массива должны вводиться с клавиатуры. Ввод организовать в цикле.

1. Объявите массивы `arr2`, `arr3`, `arr4`, `arr5` типа `int`.
2. Массив `arr2` создайте как копию всего массива `arr1`.

3. Массив `arr3` создайте как копию части массива `arr1`, задав количество копируемых элементов.
4. Массив `arr4` создайте как копию части массива `arr1`, задав диапазон.
5. Выполните сравнение массивов `arr1`, `arr2` и `arr1`, `arr3`.
6. Выполните сортировку массива `arr4`.
7. Выполните поиск значения в массивах `arr2`, `arr3`.
8. Массив `arr5` заполните указанным значением.
9. Исходные и полученные массивы вывести в виде одной строки (каждый по отдельности).

Практическое занятие 2.

Контрольная работа №3 "Типы данных"

Тема 1.4. Code conventions for the JAVA (CCJ)

Форма контроля/оценочное средство: Посещение и работа на лекционных и практических занятиях

Вопросы/Задания:

1. Посещение занятий:
 - а) посещение лекционных и практических занятий,
 - б) соблюдение дисциплины.
2. Работа на лекционных занятиях:
 - а) ведение конспекта лекций,
 - б) уровень освоения теоретического материала,
 - в) активность на лекции, умение формулировать вопросы лектору.
3. Работа на практических занятиях:

Лабораторное занятие 1.

Практика программирования, приемы программирования.

Практическое занятие в форме круглого стола проводится с целью закрепления теоретических знаний студентов по рассматриваемой теме, развития навыков командной работы, межличностной коммуникации, принятия решений.

В процессе занятия выявляются противоречия, сравниваются разные точки зрения и выбираются приемлемые для всех участников позиции и решения по определению грамотной практики программирования и выбора приемов.

Применение технологии и методов обработки исключений в процессе разработки программного средства. Анализ примеров:

- неверное преобразование строки в числовой формат
- попытка индексации вне границ массива
- деление на нуль
- методу передано неверное значение аргумента (например, отрицательное, когда метод предполагает задание положительных значений)

Пример. Написать программу, содержащую обработку исключений, в которой одному `try` соответствует сразу несколько блоков `catch` с разными классами исключений.

Лабораторное занятие 2.

Программирование согласно правилам ССJ.

Пример 1. Даны действительные числа x , y , z , вводимые с клавиатуры. Получить $\max(x + y + z, xuz)$. Создать программу с учетом:

- правильной организации файла;
- правильной структуры класса;
- правильной расстановки отступов и переносов;
- можно использовать конструкции ветвления;
- в случае, если считано не действительное число, должно выводиться сообщение «Неверный тип данных!».

Пример 2. Написать программу, которая создает три массива размерностью n . Размерность массивов n должна вводиться с клавиатуры. Заполнение массивов выполнить случайными

числами (массив №1 – случайными значениями типа int от -20 до 20 включительно, массив №2 – значениями типа int от 0 до 50 включительно, массив №3 – значениями типа int от -5 до 5 включительно). Для этого использовать генератор случайных чисел Random. Заполнение и вывод массивов организовать в цикле.

Напечатать те элементы массива №1, которые являются положительными числами.

Напечатать те элементы массива №2, которые делятся на 3 без остатка (в случае, если массив окажется пустым, вывести сообщение об этом).

Напечатать те элементы массива №3, индексы которых являются степенями двойки (1, 2, 4, 8, 16, ...).

Создать программу с учетом:

- правильной организации файла;
- правильной структуры класса;
- правильной расстановки отступов и переносов;
- можно использовать конструкции ветвления, цикла for.

Практическое занятие 1.

Контрольная работа №4 "СС1".

Тема 1.5. Классы, библиотеки, интерфейсы, пакеты

Форма контроля/оценочное средство: Посещение и работа на лекционных и практических занятиях

Вопросы/Задания:

1. Посещение занятий:

- а) посещение лекционных и практических занятий,
- б) соблюдение дисциплины.

2. Работа на лекционных занятиях:

- а) ведение конспекта лекций,
- б) уровень освоения теоретического материала,
- в) активность на лекции, умение формулировать вопросы лектору.

3. Работа на практических занятиях:

Лабораторное занятие 1.

Создание классов, содержащих статические и нестатические методы.

Пример 1. Создайте класс `Chislo`, содержащий статический метод `get()` – проверяет число: положительное, отрицательное или ноль. Метод `get()` имеет параметр `x`.

В основной программе (`Program5151.java`) создайте объект `сканер` (с помощью которого должно считываться значение `x` с консоли). В случае, если считано не действительное число, должно выводиться сообщение «Неверный тип данных!». Обработку исключения выполнить в основной программе. Объявление исключения выполнить в секции `throws` метода `get()`.

Программа вызовом метода `get()` выводит ответ, к какому виду относится число.

Пример 2. Создайте класс `Arithmetic`, содержащий нестатические методы, описывающие основные арифметические действия над целочисленными значениями. Предусмотреть ситуации, что методы могут выбрасывать исключения.

Все исключения описать в объявлении метода, обработку исключения передать для обработки далее коду, вызвавшему метод.

Практическое занятие 1.

Конструкторы. Объявление конструкторов. Особенности конструкторов. Логика работы конструкторов. Ключевые слова `super` и `this`.

Лабораторное занятие 2.

Создание программ, содержащих конструкторы.

Пример 1. Создать общедоступные классы, спецификации которых приведены ниже. Определить общедоступный конструктор для каждого класса. Структуру данных описать в отдельном классе. В основной программе выполнить реализацию и вывести все данные на

экран. Имена параметров и полей класса должны быть одинаковыми.

1. Patient: id, Фамилия, Имя, Отчество, Адрес, Телефон, Номер медицинской карты, диагноз.

2. Student: id, Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Специальность, Курс, Группа, Оценки за сессию.

Оценки за сессию – тип данных массив.

Пример 2. Создать общедоступные классы, спецификации которых приведены ниже. Определить общедоступный конструктор для каждого класса. Структуру данных описать в отдельном классе. В основной программе выполнить реализацию и вывести все данные на экран.

1. Car: id, Марка, Модель, Год выпуска, Цвет, Цена, Регистрационный номер.

2. Customer: id, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счета.

Пример 3. Создать общедоступный класс Vector. Поля класса: vx (private int), vy (private int), length (protected double).

- vx, vy – это координаты вектора;

- length – это длина вектора.

Определить общедоступные конструкторы для класса:

Конструктор 1: два аргумента – координаты вектора: int x, int y. Инициализировать поля класса. При этом для инициализации length (длина) использовать формулу вычисления длины вектора по его координатам.

Конструктор 2: четыре аргумента – координаты точек – начала и конца вектора: int x1, int y1, int x2, int y2. Инициализировать поля класса. При этом для инициализации length (длина) использовать формулу вычисления длины вектора также по его координатам (т.е. инициализаторы для length в обоих конструкторах одинаковы).

Не использовать this при инициализации переменных и ссылки на конструктор.

В основной программе выполнить реализацию (создать два экземпляра на основе двух разных конструкторов, значения параметров вводятся с клавиатуры) и вывести все данные на экран.

Лабораторное занятие 3.

Создание программ, содержащих блоки инициализации.

Пример 1. Создайте два класса: SomeClass01, SomeClass02. Выполните инициализацию нестатических полей в месте объявления (по одному полю в каждом классе). Значения полей – это значения, возвращаемые методами. В методах присутствуют стандартные потоки вывода на экран.

В основной программе выполните реализацию: в программе должны быть созданы экземпляры классов и также стандартные потоки вывода на экран.

По результатам выполнения программы и порядку выводимых строк на экран, должно быть видно, что «инициализация происходит во время загрузки класса».

Пример 2. Создайте два класса: SomeClass03, SomeClass04. Выполните инициализацию статических полей в месте объявления (по одному полю в каждом классе). Значения полей – это значения, возвращаемые методами. В методах присутствуют стандартные потоки вывода на экран.

В основной программе выполните реализацию: в программе должны быть созданы методы, которые приводят к загрузке класса и инициализации его статической части. По результатам выполнения программы и порядку выводимых строк на экран, должно быть видно, что «инициализация происходит во время загрузки класса».

Для классов SomeClass03, SomeClass04 необходимо вывести: имена самих классов и суперклассов.

Практическое занятие 2.

Статические инициализаторы. Нестатические инициализаторы (объектные). Порядок выполнения операторов при наличии блоков инициализации главного класса приложения (содержащего метод main).

Лабораторное занятие 4.

Создание программ с инициализаторами.

Пример 1. Создать класс `MaxXYZ`. Объявить статические поля: `x`, `y`, `z`. Выполнить инициализацию `x`, `y` в местах объявления полей: случайное целочисленное значение из отрезка $[1, 10]$. Используя блок статической инициализации выполнить инициализацию `z`: присвоить наибольшее значение из двух – `x`, `y`. Если `x = y`, то `z = 1`.

В основной программе найти округленное значение арифметического корня из суммы `x`, `y`, `z`.

Пример 2. Создать класс `Array5185`. Объявить статический массив символов размерностью 20. Выполнить инициализацию элементов массива с помощью статического инициализатора: в цикле присвоить случайные символьные значения из алфавита $[a, я]$.

В основной программе вывести исходный массив и отсортированный по алфавиту.

Практическое занятие 3.

Контрольная работа по первой части темы №5 "Классы, библиотеки, интерфейсы, пакеты"

Раздел 2. Часть 2

Контролируемые ИДК: УК-2.1 УК-2.2 ОПК-2.1 ОПК-7.1 ОПК-2.2 ОПК-7.2

Тема 2.1. Классы, библиотеки, интерфейсы, пакеты

Форма контроля/оценочное средство: Посещение и работа на лекционных и практических занятиях

Вопросы/Задания:

1. Посещение занятий:

- а) посещение лекционных и практических занятий,
- б) соблюдение дисциплины.

2. Работа на лекционных занятиях:

- а) ведение конспекта лекций,
- б) уровень освоения теоретического материала,
- в) активность на лекции, умение формулировать вопросы лектору.

3. Работа на практических занятиях:

Лабораторное занятие 1.

Создание программ и абстрактных классов.

Пример 1. Создайте абстрактный класс `MathAbstract`. В нем создайте два метода:

- `operMath()` без реализации с параметрами `int a`, `int b`;
- `show()` без параметров, реализующий вывод строки на экран «Иллюстрация работы с абстрактным классом: арифметические операции».

Создайте неабстрактные классы `MathReal01` – `MathReal05` – производные от `MathAbstract`. В них выполните реализации метода `operMath()`. Реализации соответствуют выполнению основных арифметических операций (сложение, вычитание, умножение, деление, возведение в степень).

В основной программе выполните реализацию: вызов всех методов класса `MathAbstract` и выполнение соответствующих операций. Значения `a`, `b` вводятся с клавиатуры.

Пример 2. Создайте абстрактный класс `BoolAbstract`. В нем создайте два метода:

- `bool()` без реализации с параметрами `boolean a`, `boolean b`;
- `show()` без параметров, реализующий вывод строки на экран «Иллюстрация работы с абстрактным классом: булевы операции».

Создайте неабстрактные классы `BoolReal01` – `BoolReal04` – производные от `BoolAbstract`. В них выполните реализации метода `bool()`. Реализации соответствуют вычислению булевых выражений.

В основной программе выполните реализацию: вызов методов класса `BoolAbstract` и вычисление соответствующих выражений. Значения `a`, `b` вводятся с клавиатуры.

Лабораторное занятие 2.

Создание интерфейсов. Разграничение доступа в Java.

Примеры.

Структура данных:

Создать классы, спецификации которых приведены ниже. Определить конструктор и методы setТип(), getТип(), printInfo().

Реализация:

Реализовать класс в основной программе.

Определить метод в классе основной программы, создающий массив объектов.

Задать критерий (критерии) выбора данных и вывести эти данные на консоль.

Продемонстрировать подход, в котором реализуется инкапсуляция и обеспечивается модульность.

Все классы должны находиться в пакете PackageClass, реализации – в пакете PackageMain.

Спецификации:

1. Student: id, Фамилия, Имя, Отчество, Адрес, Телефон, Специальность, Курс, Группа, Оценки за сессию.

Создать массив объектов. Вывести:

- 1) список студентов заданной специальности
- 2) список студентов, имеющих неудовлетворительные оценки
- 3) список студентов заданной учебной группы
- 4) список студентов, чей средний балл меньше заданного

2. Patient: id, Фамилия, Имя, Отчество, Адрес, Телефон, Номер медицинской карты, диагноз.

Создать массив объектов. Вывести:

- 1) список пациентов, имеющих данный диагноз;
- 2) список пациентов, номер медицинской карты у которых находится в заданном интервале.

3. Book: id, Название, Автор(ы), Издательство, Год издания, Количество страниц, Цена, Переплет.

Создать массив объектов. Вывести:

- 1) список книг заданного автора;
- 2) список книг, выпущенных заданным издательством;
- 3) список книг, выпущенных после заданного года.

4. House: id, Номер квартиры, Площадь, Этаж, Количество комнат, Улица, Тип здания, Срок эксплуатации.

Создать массив объектов. Вывести:

- 1) список квартир, имеющих заданное число комнат;
- 2) список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в заданном промежутке;
- 3) список квартир, имеющих площадь, превосходящую заданную.

5. Car: id, Марка, Модель, Год выпуска, Цвет, Цена, Регистрационный номер.

Создать массив объектов. Вывести:

- 1) список автомобилей заданной марки;
- 2) список автомобилей заданной модели, которые эксплуатируются больше n лет;
- 3) список автомобилей заданного года выпуска, цена которых больше указанной.

Тема 2.2. Java API

Форма контроля/оценочное средство: Посещение и работа на лекционных и практических занятиях

Вопросы/Задания:

1. Посещение занятий:

- а) посещение лекционных и практических занятий,
- б) соблюдение дисциплины.

2. Работа на лекционных занятиях:

- а) ведение конспекта лекций,
- б) уровень освоения теоретического материала,
- в) активность на лекции, умение формулировать вопросы лектору.

3. Работа на практических занятиях:

Лабораторное занятие 1.

Создание web-страниц. Тэг APPLET. Апплеты: жизненный цикл, создание.

Лабораторное занятие 2.

Изучение дополнительных методов. Класс FontMetrics. Технология работы со шрифтами. Методы класса FontMetrics.

Лабораторное занятие 3.

Подклассы класса Component. Абстрактные классы AWT. Подклассы класса Container. Создание апплетов, используя подклассы класса Component.

Лабораторное занятие 4.

Технология создания меню. Лабораторная работа проводится в интерактивной форме (работа в малых группах), что позволяет развивать навыки межличностной коммуникации, командной работы и принятия решений.

Лабораторное занятие 5.

Схема MVC в компонентах Swing. Создание окна JFrame. Панель содержимого, Container, JPanel.

Пример. Используя менеджеры размещения, разработать форму расположения элементов игры «Пятнашки».

Лабораторное занятие 6.

Особенности выравнивания элементов. Ручное и автоматическое размещение элементов. Пример. Используя менеджеры размещения, разработать форму регистрации пользователя, содержащую различные элементы управления.

Лабораторное занятие 7.

Создание GUI-приложения "Калькулятор".

Тема 2.3. Дженирики и коллекции

Форма контроля/оценочное средство: Посещение и работа на лекционных и практических занятиях

Вопросы/Задания:

1. Посещение занятий:

- а) посещение лекционных и практических занятий,
- б) соблюдение дисциплины.

2. Работа на лекционных занятиях:

- а) ведение конспекта лекций,
- б) уровень освоения теоретического материала,
- в) активность на лекции, умение формулировать вопросы лектору.

3. Работа на практических занятиях:

Лабораторное занятие 1.

Дженерик-классы. Дженерик-методы.

Лабораторное занятие 2.

Создание приложения, используя коллекции.

Лабораторное занятие 3.

Использование объектно-ориентированного дизайна (ООД) классов в разработке приложений.

Тема 2.4. Разработка приложения с GUI-интерфейсом

Форма контроля/оценочное средство: Посещение и работа на лекционных и практических занятиях

Вопросы/Задания:

1. Посещение занятий:

- а) посещение лекционных и практических занятий,
- б) соблюдение дисциплины.

2. Работа на лекционных занятиях:

- а) ведение конспекта лекций,
- б) уровень освоения теоретического материала,
- в) активность на лекции, умение формулировать вопросы лектору.

3. Работа на практических занятиях:

Лабораторное занятие 1.

Создание приложения с GUI-интерфейсом для автоматизации бизнес-процессов, решения прикладных задач различных классов. Варианты бизнес-процессов:

- учет сервисного обслуживания оборудования;
- учет продления лицензий на ПО;
- учет заявок на профилактические работы;
- расчет расхода топлива на транспортные услуги;
- расчет стоимости оказания услуг и т.п.

Этапы выполнения задания:

1. Постановка задачи, выбор направления автоматизации
2. Формулирование цели и задач проекта по созданию программы
3. Разработки плана и целевых этапов создания приложения
4. Составление алгоритма. Анализ альтернативных вариантов реализации алгоритма для достижения намеченных результатов и реализации программного решения
5. Выбор программного обеспечения для разработки приложения
6. Разработка приложения.

9. Оценочные материалы промежуточной аттестации

Очная форма обучения, Пятый семестр, Зачет

Контролируемые ИДК: УК-2.1 УК-2.2 ОПК-2.1 ОПК-7.1 ОПК-2.2 ОПК-7.2

Вопросы/Задания:

1. Работа с тестовыми заданиями

Тестовые задания представлены в приложении 7

2. Выполнение итоговой работы

Примерный перечень вопросов к зачету по дисциплине представлен в приложении 9.

Типовые практические задания на этапе промежуточной аттестации

Ход работы

1. Выберите вариант практического задания. Номер варианта и перечень заданий выбираются по таблице 1 (Приложение 8).
2. Сформулируйте цели и задачи проекта по созданию программ.
3. Выберите инструментальные средства разработки алгоритмов и программ. Настройте программную среду для разработки консольных приложений.
4. Разработайте алгоритмы для решения задач. Проанализируйте альтернативные варианты реализации алгоритмов.
5. Разработайте консольные приложения в выбранной инструментальной среде. Выполните их тестирование и отладку.

Основные требования к отчету

Все файлы (листинги программ, скомпилированные файлы и др.) выполненного задания, включая отчет, должны быть размещены в папке, имя которой содержит название дисциплины и фамилию студента. В систему дистанционного обучения должен быть загружен

архив этой папки (.zip или 7z).

Например: ООП_ИвановИИ.zip

В файле – отчете (ООП_ИвановИИ.docx) должны располагаться:

- 1) титульный лист;
- 2) текст заданий и листинги с комментариями.

Очная форма обучения, Шестой семестр, Экзамен

Контролируемые ИДК: УК-2.1 УК-2.2 ОПК-2.1 ОПК-7.1 ОПК-2.2 ОПК-7.2

Вопросы/Задания:

1. Работа с тестовыми заданиями

Тестовые задания представлены в приложении 7.

2. Выполнение итоговой работы

Примерный перечень вопросов к экзамену по дисциплине представлен в приложении 9.

Типовые практические задания на этапе промежуточной аттестации

Ход работы

1. Выберите вариант практического задания. Изучите текст заданий. При необходимости выполните уточнение заданий. (Приложение 8)
2. Выполните анализ предметной области. Сформулируйте цели и задачи проекта по созданию программ.
3. Выберите инструментальные средства разработки программы. Настройте программную среду для разработки приложений с GUI-интерфейсом.
4. Разработайте алгоритмы для решения задач. Проанализируйте альтернативные варианты реализации алгоритмов.
5. Разработайте приложения с GUI-интерфейсом в выбранной инструментальной среде. Выполните их тестирование и отладку.
6. Предложите вариант реализации приложения с базой данных. Как будет выглядеть схема данных БД?

Основные требования к отчету

Все файлы (листинги программ, скомпилированные файлы и др.) выполненного задания, включая отчет, должны быть размещены в папке, имя которой содержит название дисциплины и фамилию студента. В систему дистанционного обучения должен быть загружен архив этой папки (.zip или 7z).

Например: ООП_ИвановИИ.zip

В файле – отчете (ООП_ИвановИИ.docx) должны располагаться:

- 1) титульный лист;
- 2) текст заданий и листинги с комментариями;
- 3) скриншоты форм созданного приложения.

Очно-заочная форма обучения, Шестой семестр, Зачет

Контролируемые ИДК: УК-2.1 УК-2.2 ОПК-2.1 ОПК-7.1 ОПК-2.2 ОПК-7.2

Вопросы/Задания:

1. Работа с тестовыми заданиями

Тестовые задания представлены в приложении 7

2. Выполнение итоговой работы

Примерный перечень вопросов к зачету по дисциплине представлен в приложении 9.

Типовые практические задания на этапе промежуточной аттестации

Ход работы

1. Выберите вариант практического задания. Номер варианта и перечень заданий выбираются по таблице 1 (Приложение 8).
2. Сформулируйте цели и задачи проекта по созданию программ.
3. Выберите инструментальные средства разработки алгоритмов и программ. Настройте программную среду для разработки консольных приложений.
4. Разработайте алгоритмы для решения задач. Проанализируйте альтернативные варианты

реализации алгоритмов.

5. Разработайте консольные приложения в выбранной инструментальной среде. Выполните их тестирование и отладку.

Основные требования к отчету

Все файлы (листинги программ, скомпилированные файлы и др.) выполненного задания, включая отчет, должны быть размещены в папке, имя которой содержит название дисциплины и фамилию студента. В систему дистанционного обучения должен быть загружен архив этой папки (.zip или 7z.).

Например: ООП_ИвановИИ.zip

В файле – отчете (ООП_ИвановИИ.docx) должны располагаться:

- 1) титульный лист;
- 2) текст заданий и листинги с комментариями.

Очно-заочная форма обучения, Седьмой семестр, Экзамен

Контролируемые ИДК: УК-2.1 УК-2.2 ОПК-2.1 ОПК-7.1 ОПК-2.2 ОПК-7.2

Вопросы/Задания:

1. Работа с тестовыми заданиями

Тестовые задания представлены в приложении 7.

2. Выполнение итоговой работы

Примерный перечень вопросов к экзамену по дисциплине представлен в приложении 9.

Типовые практические задания на этапе промежуточной аттестации

Ход работы

1. Выберите вариант практического задания. Изучите текст заданий. При необходимости выполните уточнение заданий. (Приложение 8)
2. Выполните анализ предметной области. Сформулируйте цели и задачи проекта по созданию программ.
3. Выберите инструментальные средства разработки программы. Настройте программную среду для разработки приложений с GUI-интерфейсом.
4. Разработайте алгоритмы для решения задач. Проанализируйте альтернативные варианты реализации алгоритмов.
5. Разработайте приложения с GUI-интерфейсом в выбранной инструментальной среде. Выполните их тестирование и отладку.
6. Предложите вариант реализации приложения с базой данных. Как будет выглядеть схема данных БД?

Основные требования к отчету

Все файлы (листинги программ, скомпилированные файлы и др.) выполненного задания, включая отчет, должны быть размещены в папке, имя которой содержит название дисциплины и фамилию студента. В систему дистанционного обучения должен быть загружен архив этой папки (.zip или 7z.).

Например: ООП_ИвановИИ.zip

В файле – отчете (ООП_ИвановИИ.docx) должны располагаться:

- 1) титульный лист;
- 2) текст заданий и листинги с комментариями;
- 3) скриншоты форм созданного приложения.

Заочная форма обучения, Шестой семестр, Зачет

Контролируемые ИДК: УК-2.1 УК-2.2 ОПК-2.1 ОПК-7.1 ОПК-2.2 ОПК-7.2

Вопросы/Задания:

1. Работа с тестовыми заданиями

Тестовые задания представлены в приложении 7

2. Выполнение итоговой работы

Примерный перечень вопросов к зачету по дисциплине представлен в приложении 9.

Типовые практические задания на этапе промежуточной аттестации

Ход работы

1. Выберите вариант практического задания. Номер варианта и перечень заданий выбираются по таблице 1 (Приложение 8).
2. Сформулируйте цели и задачи проекта по созданию программ.
3. Выберите инструментальные средства разработки алгоритмов и программ. Настройте программную среду для разработки консольных приложений.
4. Разработайте алгоритмы для решения задач. Проанализируйте альтернативные варианты реализации алгоритмов.
5. Разработайте консольные приложения в выбранной инструментальной среде. Выполните их тестирование и отладку.

Основные требования к отчету

Все файлы (листинги программ, скомпилированные файлы и др.) выполненного задания, включая отчет, должны быть размещены в папке, имя которой содержит название дисциплины и фамилию студента. В систему дистанционного обучения должен быть загружен архив этой папки (.zip или 7z.).

Например: ООП_ИвановИИ.zip

В файле – отчете (ООП_ИвановИИ.docx) должны располагаться:

- 1) титульный лист;
- 2) текст заданий и листинги с комментариями.

Заочная форма обучения, Седьмой семестр, Экзамен

Контролируемые ИДК: УК-2.1 УК-2.2 ОПК-2.1 ОПК-7.1 ОПК-2.2 ОПК-7.2

Вопросы/Задания:

1. Работа с тестовыми заданиями

Тестовые задания представлены в приложении 7.

2. Выполнение итоговой работы

Примерный перечень вопросов к экзамену по дисциплине представлен в приложении 9.

Типовые практические задания на этапе промежуточной аттестации

Ход работы

1. Выберите вариант практического задания. Изучите текст заданий. При необходимости выполните уточнение заданий. (Приложение 8)
2. Выполните анализ предметной области. Сформулируйте цели и задачи проекта по созданию программ.
3. Выберите инструментальные средства разработки программы. Настройте программную среду для разработки приложений с GUI-интерфейсом.
4. Разработайте алгоритмы для решения задач. Проанализируйте альтернативные варианты реализации алгоритмов.
5. Разработайте приложения с GUI-интерфейсом в выбранной инструментальной среде. Выполните их тестирование и отладку.
6. Предложите вариант реализации приложения с базой данных. Как будет выглядеть схема данных БД?

Основные требования к отчету

Все файлы (листинги программ, скомпилированные файлы и др.) выполненного задания, включая отчет, должны быть размещены в папке, имя которой содержит название дисциплины и фамилию студента. В систему дистанционного обучения должен быть загружен архив этой папки (.zip или 7z.).

Например: ООП_ИвановИИ.zip

В файле – отчете (ООП_ИвановИИ.docx) должны располагаться:

- 1) титульный лист;
- 2) текст заданий и листинги с комментариями;
- 3) скриншоты форм созданного приложения.

10. Материально-техническое и учебно-методическое обеспечение дисциплины

10.1. Перечень основной и дополнительной учебной литературы

Основная литература

1. Объектно-ориентированное программирование: учебное пособие: учебное пособие / П. П. Степанов, А. А. Кабанов, В. А. Никонов, Т. С. Павлюченко; ред. К. В. Обухова. - Омск: Омский государственный технический университет (ОмГТУ), 2021. - 112 с. - 978-5-8149-3301-0 (ч. 1). – ISBN 978-5-8149-3300-3. - Текст: электронный. // Директ-Медиа: [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=700657> (дата обращения: 26.09.2024). - Режим доступа: по подписке

Дополнительная литература

1. Карякин, М. И. Технологии программирования и компьютерный практикум на языке Python: учебное пособие: учебное пособие / М. И. Карякин, К. А. Ватульян, Р. М. Мнухин. - Ростов-на-Дону, Таганрог: Южный федеральный университет, 2022. - 244 с. - 978-5-9275-4108-9. - Текст: электронный. // Директ-Медиа: [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=698687> (дата обращения: 26.09.2024). - Режим доступа: по подписке

2. Программирование: учебное пособие: учебное пособие / О. Ю. Горбадей, О. П. Рябычина, А. Н. Балаш, Н. И. Янович. - Минск: РИПО, 2023. - 300 с. - 978-985-895-148-1. - Текст: электронный. // Директ-Медиа: [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=717782> (дата обращения: 26.09.2024). - Режим доступа: по подписке

3. Шкодина, Т. А. Алгоритмы и структуры данных в Python: лабораторный практикум: учебное пособие: учебное пособие / Т. А. Шкодина. - Ростов-на-Дону: Издательско-полиграфический комплекс РГЭУ (РИНХ), 2022. - 80 с. - 978-5-7972-3037-3. - Текст: электронный. // Директ-Медиа: [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=704757> (дата обращения: 26.09.2024). - Режим доступа: по подписке

4. Программирование, тестирование, проектирование, нейросети, технологии аппаратно-программных средств (практические задания и способы их решения): учебник: учебник / С. В. Веретехина, К. С. Кармицкий, Д. Д. Лукашин [и др.] - Москва: Директ-Медиа, 2022. - 144 с. - 978-5-4499-3321-8. - Текст: электронный. // Директ-Медиа: [сайт]. - URL: <https://doi.org/10.23681/694782> (дата обращения: 26.09.2024). - Режим доступа: по подписке

5. Проскураков, А. В. Качество и тестирование программного обеспечения. Метрология программного обеспечения: учебное пособие: учебное пособие / А. В. Проскураков. - Ростов-на-Дону, Таганрог: Южный федеральный университет, 2022. - 199 с. - 978-5-9275-4044-0. - Текст: электронный. // Директ-Медиа: [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=698742> (дата обращения: 26.09.2024). - Режим доступа: по подписке

6. Златопольский, Д. М. Программирование: типовые задачи, алгоритмы, методы: учебное пособие: учебное пособие / Д. М. Златопольский. - 4-е изд. (эл.) - Москва: Лаборатория знаний, 2020. - 226 с. - 978-5-00101-789-9. - Текст: электронный. // Директ-Медиа: [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=222873> (дата обращения: 26.09.2024). - Режим доступа: по подписке

7. Калитвин, В. А. Введение в программирование на Python: учебное пособие: учебное пособие / В. А. Калитвин. - Липецк: Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2023. - 85 с. - 978-5-907655-86-7. - Текст: электронный. // Директ-Медиа: [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=714538> (дата обращения: 26.09.2024). - Режим доступа: по подписке

10.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся

Профессиональные базы данных

1. <http://www.ebiblioteka.ru> - Базы данных East View
2. <https://academic.microsoft.com> - Международная научная реферативная база данных
3. <https://scholar.google.ru> - Международная научная реферативная база данных

Ресурсы «Интернет»

1. <http://www.sibit.sano.ru> - Официальный сайт образовательной организации
2. <http://do.sano.ru> - Система дистанционного обучения Moodle (СДО Moodle)
3. <http://www.edu.ru> - Федеральный портал «Российское образование»
4. <http://lib.ru> - Электронная библиотека Максима Мошкова
5. www.ucheba.com - Образовательный портал «Учёба»

10.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине

При подготовке и проведении учебных занятий по дисциплине студентами и преподавателями используются следующие современные профессиональные базы данных и информационно-справочные системы:

1. Электронная библиотечная система «Университетская библиотека онлайн» (<http://www.biblioclub.ru>).
2. Интегрированная библиотечно-информационная система ИРБИС64 (<http://lib.sano.ru>).
3. справочно-правовая система КонсультантПлюс.
4. Электронная справочная система ГИС Омск.

10.4. Специальные помещения, лаборатории и лабораторное оборудование

Институт располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Для проведения учебных занятий по дисциплине используются следующие помещения, оснащенные оборудованием и техническими средствами обучения:

Для лекций, семинаров (практических), групповых, индивидуальных консультаций, текущего контроля, промежуточной аттестации, ГИА

Мультимедийная учебная аудитория № 210

Перечень оборудования

- Аудиоколонка - 5 шт.
- Доска маркерная - 1 шт.
- Компьютер с выходом в Интернет - 1 шт.
- Проектор - 1 шт.
- Стол - 37 шт.
- Стол преподавателя - 1 шт.
- Стул - 74 шт.
- Стул преподавателя - 1 шт.
- Трибуна - 1 шт.
- Экран - 1 шт.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

- Adobe Acrobat Reader
- Kaspersky Endpoint Security для Windows

Microsoft Office 2007 standart Win32 Russian
Microsoft Windows XP Professional Russian

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Consultant Plus
2GIS

Мультимедийная учебная аудитория № 211

Перечень оборудования

Аудиоколонка - 5 шт.
Доска маркерная - 1 шт.
Компьютер с выходом в Интернет - 1 шт.
Проектор - 1 шт.
Стол - 27 шт.
Стол преподавателя - 1 шт.
Стул - 54 шт.
Стул преподавателя - 1 шт.
Трибуна - 1 шт.
Экран - 1 шт.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

Adobe Acrobat Reader
Kaspersky Endpoint Security для Windows
Microsoft Office 2007 standart Win32 Russian
Microsoft Windows XP Professional Russian

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Consultant Plus
2GIS

Мультимедийная учебная аудитория № 304

Перечень оборудования

Аудиоколонка - 2 шт.
Доска маркерная - 1 шт.
Компьютер с выходом в Интернет - 1 шт.
Проектор - 0 шт.
Стол - 18 шт.
Стол преподавателя - 1 шт.
Стул - 36 шт.
Стул преподавателя - 1 шт.
Тематические иллюстрации - 0 шт.
Трибуна - 1 шт.
Учебно-наглядные пособия - 0 шт.
Экран - 0 шт.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

Adobe Acrobat Reader
Kaspersky Endpoint Security для Windows
Microsoft Office 2007 standart Win32 Russian

Microsoft Windows 10

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Consultant Plus
2GIS

Мультимедийная учебная аудитория № 312

Перечень оборудования

Аудиоколонка - 2 шт.
Компьютер с выходом в Интернет - 1 шт.
Проектор - 1 шт.
Тематические иллюстрации - 0 шт.
Учебно-наглядные пособия - 0 шт.
Экран - 1 шт.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

Adobe Acrobat Reader
Kaspersky Endpoint Security для Windows
Microsoft Office 2007 standart Win32 Russian
Microsoft Windows XP Professional Russian

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Consultant Plus
2GIS

Мультимедийная учебная аудитория № 422

Перечень оборудования

Аудиоколонка - 2 шт.
Доска маркерная - 1 шт.
Интерактивная доска - 1 шт.
Компьютер с выходом в Интернет - 1 шт.
Стол - 13 шт.
Стол преподавателя - 1 шт.
Стул - 26 шт.
Стул преподавателя - 1 шт.
Трибуна - 1 шт.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

Adobe Acrobat Reader
Kaspersky Endpoint Security для Windows
Microsoft Office 2007 standart Win32 Russian
Microsoft Windows 8 Professional Russian

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Consultant Plus
2GIS

Для семинаров (практических, лабораторных), консультаций, текущего контроля,

промежуточной аттестации, ГИА, НИР, курсового проектирования

Лаборатория иностранных языков и информационных дисциплин № 401

Перечень оборудования

Доска маркерная - 1 шт.
Интерактивная доска - 1 шт.
Информационная доска - 1 шт.
Лингафонное оборудование - 0 шт.
Мультимедиапроектор - 1 шт.
Наушники с микрофоном - 10 шт.
Персональный компьютер - 11 шт.
Стол - 8 шт.
Стол преподавателя - 1 шт.
Стул - 16 шт.
Стул преподавателя - 1 шт.
Тематические иллюстрации - 0 шт.
Учебно-наглядные пособия - 0 шт.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

Adobe Acrobat Reader
Kaspersky Endpoint Security для Windows
Microsoft Office стандартный 2016
Microsoft Access 2016
Joy Class
NetBeansIDE
Microsoft Visual Studio 2017 CE (C#, C++)
Microsoft Visual Studio 2010 Express
Microsoft Visual Studio Community
Microsoft SQL 2010 Express
Notepad ++
MySQL
OracleSQLDeveloper
Microsoft SOAPToolkit
CADE
Denwer 3 webserver
Dev-C++
IDEEclipse
JDK 6
Freepascal
Lazarus
Geany
JavaDevelopmentKit
TheRProject
NetBeansIDE8
StarUML 5.0.2
EViews 9 StudentVersionLite
Gretl
Matrixer
Maxima
Xmind
BPWIN
IrfanView
SMARTBoard

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Consultant Plus
2GIS

Лаборатория экономических и информационных дисциплин № 402

Перечень оборудования

Доска маркерная - 1 шт.
Персональный компьютер - 10 шт.
Стол - 13 шт.
Стол преподавателя - 1 шт.
Стул - 16 шт.
Стул преподавателя - 1 шт.
Тематические иллюстрации - 0 шт.
Учебно-наглядные пособия - 0 шт.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

Adobe Acrobat Reader
Kaspersky Endpoint Security для Windows
NetBeansIDE
Microsoft Visual Studio 2017 CE (C#, C++)
Microsoft Visual Studio 2010 Express
Microsoft Visual Studio Community
Microsoft SQL 2010 Express
Notepad ++
MySQL
OracleSQLDeveloper
Microsoft SOAPToolkit
CADE
Denwer 3 webserver
Dev-C++
IIDEclipse
JDK 6
Freepascal
Lazarus
Geany
JavaDevelopmentKit
TheRProject
NetBeansIDE8
StarUML 5.0.2
EViews 9 StudentVersionLite
Gretl
Matrixer
Maxima
Xmind
BPWIN
IrfanView
SMARTBoard
Gimp
Java 8 Update 381 (64-bit)
Microsoft Office 2013 Professional Plus Win32 Russian
1С Предприятие 8.2. Комплект для обучения в высших и средних учебных заведениях

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Consultant Plus
2GIS

Лаборатория иностранных языков и информационных дисциплин № 403

Перечень оборудования

Доска маркерная - 1 шт.
Лингафонное оборудование - 0 шт.
Наушники с микрофоном - 10 шт.
Персональный компьютер - 11 шт.
Стол - 9 шт.
Стол преподавателя - 1 шт.
Стул - 21 шт.
Стул преподавателя - 1 шт.
Тематические иллюстрации - 0 шт.
Техническое оснащение (монитор) - 2 шт.
Учебно-наглядные пособия - 0 шт.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

Adobe Acrobat Reader
Kaspersky Endpoint Security для Windows
Microsoft Office 2007 standart Win32 Russian
NetBeansIDE
Microsoft Visual Studio 2017 CE (C#, C++)
Microsoft Visual Studio 2010 Express
Microsoft Visual Studio Community
Microsoft SQL 2010 Express
Notepad ++
MySQL
OracleSQLDeveloper
Microsoft SOAPToolkit
CADE
Denwer 3 webserver
Dev-C++
IIDEclipse
JDK 6
Freepascal
Geany
JavaDevelopmentKit
TheRProject
NetBeansIDE8
StarUML 5.0.2
EViews 9 StudentVersionLite
Gretl
Matrixer
Maxima
Xmind
BPWIN
IrfanView

NetClass
Microsoft Windows XP Professional Russian
CorelDRAW Graphics Suite X4
NetClass PRO
Gimp

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Consultant Plus
2GIS

Для лекций, семинаров (практических), групповых и индивидуальных консультаций, текущего контроля, промежуточной аттестации

Мультимедийная учебная аудитория № 305

Перечень оборудования

Аудиоколонка - 2 шт.
Доска маркерная - 1 шт.
Информационная доска - 1 шт.
Компьютер с выходом в Интернет - 1 шт.
Круглый стол - 3 шт.
Ноутбук DELL - 8 шт.
Ноутбук HP - 2 шт.
Персональный компьютер - 1 шт.
Проектор - 1 шт.
Стеллаж - 2 шт.
Стол одноместный - 10 шт.
Стол преподавателя - 1 шт.
Стул - 27 шт.
Стул преподавателя - 1 шт.
Трибуна - 1 шт.
Экран - 1 шт.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

MariaDB 10.11 (x64)
Microsoft Office 2016 standart Win64 Russian
Adobe Acrobat Reader
Kaspersky Endpoint Security для Windows

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Consultant Plus
2GIS

Помещение для хранения и профилактического обслуживания учебного, компьютерного оборудования и хранения элементов мультимедийных лабораторий

Специальное помещение № 420

Перечень оборудования

Запасные части для компьютерного оборудования - 0 шт.
Наушники для лингафонного кабинета - 0 шт.
Паяльная станция - 1 шт.

Персональный компьютер - 4 шт.
Планшетный компьютер - 15 шт.
Сервер - 10 шт.
Стеллаж - 0 шт.
Стол - 4 шт.
Стул - 4 шт.

Перечень программного обеспечения
(обновление производится по мере появления новых версий программы)
Не используется.

Перечень информационно-справочных систем
(обновление выполняется еженедельно)
Не используется.

Помещение для хранения и профилактического обслуживания учебного оборудования

Специальное помещение № 003

Перечень оборудования

Запасные части для столов и стульев - 0 шт.
Материалы для сопровождения учебного процесса - 0 шт.
Наборы слесарных инструментов для обслуживания учебного оборудования - 0 шт.
Станок для сверления - 0 шт.
Стеллаж - 0 шт.
Угловая шлифовальная машина - 0 шт.

Перечень программного обеспечения
(обновление производится по мере появления новых версий программы)
Не используется.

Перечень информационно-справочных систем
(обновление выполняется еженедельно)
Не используется.

Для семинаров (практических, лабораторных), консультаций, текущего контроля, промежуточной аттестации, курсового проектирования

Лаборатория иностранных языков и информационных дисциплин № 412

Перечень оборудования

Компьютер с выходом в Интернет - 11 шт.
Стол - 10 шт.
Стол преподавателя - 1 шт.
Стул - 10 шт.
Стул преподавателя - 1 шт.

Перечень программного обеспечения
(обновление производится по мере появления новых версий программы)

Adobe Acrobat Reader
Kaspersky Endpoint Security для Windows
Microsoft Visual Studio 2017 CE (C#, C++)
1С Предприятие 8.2. Комплект для обучения в высших и средних учебных заведениях
Microsoft Windows 10 Professional Russian
Microsoft Office профессиональный плюс 2016

7-Zip 24.08(x64)
Far Manager 3 (x64)
Microsoft Visual Studio Code
Python Launcher
PuTTY release 0.81 (64-bit)
PostgreSQL 16
PhpStorm 2024.2.0.1
PDF24 Creator 11.18.0
PyCharm Community Edition 2022.3.1
PyCharm Community Edition 2023.2.1
PyCharm Community Edition 2023.2.3
draw.io 24.7.5

Перечень информационно-справочных систем
(обновление выполняется еженедельно)
Не используется.

11. Методические указания по освоению дисциплины (модуля)

ВИДЫ И ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Успешное освоение теоретического материала по дисциплине требует самостоятельной работы, нацеленной на усвоение лекционного теоретического материала, расширение и конкретизацию знаний по разнообразным вопросам дисциплины. Самостоятельная работа студентов предусматривает следующие виды:

1. Аудиторная самостоятельная работа студентов – выполнение на практических занятиях и лабораторных работах заданий, закрепляющих полученные теоретические знания либо расширяющие их, а также выполнение разнообразных контрольных заданий индивидуального или группового характера (подготовка устных докладов или сообщений о результатах выполнения заданий, выполнение самостоятельных проверочных работ по итогам изучения отдельных вопросов и тем дисциплины);
2. Внеаудиторная самостоятельная работа студентов – подготовка к лекционным, практическим занятиям, лабораторным работам, повторение и закрепление ранее изученного теоретического материала, конспектирование учебных пособий и периодических изданий, изучение проблем, не выносимых на лекции, написание тематических рефератов, выполнение индивидуальных практических заданий, подготовка к тестированию по дисциплине, выполнение итоговой работы.

Большое значение в преподавании дисциплины отводится самостоятельному поиску студентами информации по отдельным теоретическим и практическим вопросам и проблемам.

При планировании и организации времени для изучения дисциплины необходимо руководствоваться п. 5.1 или 5.2 рабочей программы дисциплины и обеспечить последовательное освоение теоретического материала по отдельным вопросам и темам (Приложение 3).

Наиболее целесообразен следующий порядок изучения теоретических вопросов по дисциплине:

1. Изучение справочников (словарей, энциклопедий) с целью уяснения значения основных терминов, понятий, определений;
2. Изучение учебно-методических материалов для лекционных, практических занятий, лабораторных работ;
3. Изучение рекомендуемой основной и дополнительной литературы и электронных информационных источников;
4. Изучение дополнительной литературы и электронных информационных источников, определенных в результате самостоятельного поиска информации;
5. Самостоятельная проверка степени усвоения знаний по контрольным вопросам и/или заданиям;

6. Повторное и дополнительное (углубленное) изучение рассмотренного вопроса (при необходимости).

В процессе самостоятельной работы над учебным материалом рекомендуется составить конспект, где кратко записать основные положения изучаемой темы. Переходить к следующему разделу можно после того, когда предшествующий материал понят и усвоен. В затруднительных случаях, встречающихся при изучении курса, необходимо обратиться за консультацией к преподавателю.

При изучении дисциплины не рекомендуется использовать материалы, подготовленные неизвестными авторами, размещенные на неофициальных сайтах неделового содержания. Желательно, чтобы используемые библиографические источники были изданы в последние 3-5 лет. Студенты при выполнении самостоятельной работы могут воспользоваться учебно-методическими материалами по дисциплине, представленными в электронной библиотеке института, и предназначенными для подготовки к лекционным и семинарским занятиям.

Контроль аудиторной самостоятельной работы осуществляется в форме дискуссии и собеседования. Контроль внеаудиторной самостоятельной работы студентов осуществляется в форме устного или письменного опроса. Промежуточный контроль знаний в форме экзамена осуществляется посредством письменного тестирования, включающего вопросы и задания для самостоятельного изучения.

Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Оценка компетенций на различных этапах их формирования осуществляется в соответствии с Положением о текущем контроле и промежуточной аттестации, Положением о балльной и рейтинговой системах оценивания и технологической картой дисциплины (Приложение 1-2). Показатели и критерии оценивания компетенций на этапе текущего и промежуточного контроля представлены в Приложении 4.

Промежуточная аттестация по дисциплине проводится в форме экзамена/зачета в виде выполнения тестирования и/или итоговой работы.

Итоговые задания разрабатываются по основным вопросам теоретического материала и позволяют осуществлять промежуточный контроль знаний и степени усвоения материала.

При проведении промежуточной аттестации студентов по дисциплине могут формироваться варианты тестов, относящихся ко всем темам дисциплины (Приложение 7)

Оценка знаний студентов осуществляется в соответствии с Положением о балльной и рейтинговой системах оценивания, принятой в Институте, и технологической картой дисциплины

ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ НА ЭТАПЕ ТЕКУЩЕГО КОНТРОЛЯ

1) Посещение и работа на лекционных и практических занятиях (собеседование, контрольная работа, круглый стол и дискуссия)

При преподавании дисциплины «Объектно-ориентированное программирование» применяются разнообразные образовательные технологии в зависимости от вида и целей учебных занятий.

Теоретический материал излагается на лекционных занятиях в следующих формах:

- проблемные лекции;

- лекция-беседа.

Лабораторные работы и практические занятия по дисциплине «Объектно-ориентированное программирование» ориентированы на закрепление теоретического материала, изложенного на лекционных занятиях, а также на приобретение дополнительных знаний, умений и практических навыков осуществления профессиональной деятельности посредством активизации и усиления самостоятельной деятельности обучающихся.

Лабораторные работы и практические занятия проводятся с применением активных форм обучения, к которым относятся:

- 1) устный опрос студентов с элементами беседы и дискуссии по вопросам, выносимым на практические занятия;
- 2) групповая работа студентов, предполагающая совместное обсуждение какой-либо проблемы (вопроса) и выработку единого мнения (позиции) по ней (метод группового обсуждения);
- 3) контрольная работа по отдельным вопросам, целью которой является проверка знаний студентов и уровень подготовленности для усвоения нового материала по дисциплине.

На практических занятиях оцениваются и учитываются все виды активности студентов: устные ответы, дополнения к ответам других студентов, участие в дискуссиях, работа в группах, инициативный обзор проблемного вопроса, письменная работа.

2) Письменное задание

Формируемые компетенции: УК-2, ОПК-2, ОПК-7

Цели и задачи реферата.

Целью работы является обобщение и систематизация теоретического материала в рамках исследуемой проблемы.

В процессе выполнения работы решаются следующие задачи:

1. Формирование информационной базы:

- анализ точек зрения зарубежных и отечественных специалистов;
- конспектирование и реферирование первоисточников в качестве базы для сравнения, противопоставления, обобщения;
- анализ и обоснование степени изученности исследуемой проблемы;
- подготовка библиографического списка исследования.

2. Формулировка актуальности темы:

- отражение степени важности исследуемой проблемы в современной теории и практике;
- выявление соответствия задачам теории и практики, решаемым в настоящее время;
- определение места выбранной для исследования проблемы.

3. Формулировка цели и задач работы:

- изложение того, какой конечный результат предполагается получить при проведении теоретического исследования;
- четкая формулировка цели и разделение процесса ее достижения на этапы;
- выявление особенностей решения задач (задачи - это те действия, которые необходимо предпринять для достижения поставленной в работе цели).

В результате написания реферата студент изучает и анализирует информационную базу с целью установления теоретических зависимостей, формулирует понятийный аппарат, определяет актуальность, цель и задачи работы.

Обязательными составляющими элементами реферата являются:

- титульный лист;
- содержание;
- введение;
- основное содержание, разделенное на разделы (параграфы, пункты, подпункты), расположенные и поименованные согласно плану; в них аргументировано и логично раскрывается избранная тема в соответствии с поставленной целью; обзор литературы; описание применяемых методов, инструментов, методик, процедур в рамках темы исследования; анализ примеров российского и зарубежного опыта, отражающих тему исследования и т.д..
- заключение;
- список использованных источников;
- приложения.

Требования к оформлению письменных работ представлены в Методических указаниях к содержанию, оформлению и критериям оценивания письменных, практических и курсовых работ, утвержденных решением Научно-методического совета (протокол №8 от 07.06.2018 г.). Номер темы для выполнения реферата определяется по таблице представленной в приложении 5.

3) Практическое задание

Формируемые компетенции: УК-2, ОПК-2, ОПК-7

1 часть

Практическое задание на тему «Объектно-ориентированное программирование: создание консольных программ»

Целью выполнения практического задания «Объектно-ориентированное программирование: создание программ» является закрепление навыков создания программ на объектно-ориентированном языке программирования (Java).

Примечание. Для выполнения практического задания может быть выбран другой язык ООП.

Листинг программ должен быть оформлен на листах формата А4 и в электронном виде.

Листинг в обязательном порядке должен содержать комментарии.

Номер варианта для выполнения практических заданий определяется по таблице 1 (прил. 6).

Перечень заданий приведен в прил. 6.

План выполнения кейса

1. Знакомство с задачей кейса (обзор кейса, постановка задачи, уточнение задания).
2. Выявление основной проблемы.
3. Рассмотрение альтернативных путей решения.
4. Выбор оптимального решения.
5. Реализация решения.
6. Предоставление отчета.

Постановка задачи

1. Выберите вариант практического задания. Изучите текст заданий. При необходимости выполните уточнение заданий.
2. Выявите проблемы, которые необходимо решить, выполнив задания (разрабатываемого приложения).
3. Выберите инструментальные средства разработки алгоритмов и программ. Настройте программную среду для разработки консольных приложений.
4. Разработайте алгоритмы для решения задач. Проанализируйте альтернативные варианты реализации алгоритмов.
5. Разработайте консольные приложения в выбранной инструментальной среде. Выполните их тестирование и отладку.

Основные требования к отчету

Все файлы (листинги программ, скомпилированные файлы и др.) выполненного задания, включая отчет, должны быть размещены в папке, имя которой содержит название дисциплины и фамилию студента. В систему дистанционного обучения должен быть загружен архив этой папки (.zip или 7z).

Например: ООП_ИвановИИ.zip

В файле – отчете (ООП_ИвановИИ.docx) должны располагаться:

- 1) титульный лист;
- 2) текст заданий и листинги с комментариями.

2 часть

Практическое задание на тему «Объектно-ориентированное программирование: создание программ с GUI-интерфейсом»

Целью выполнения практического задания «Объектно-ориентированное программирование: создание программ с GUI-интерфейсом» является закрепление навыков создания программ с графическим интерфейсом на объектно-ориентированном языке программирования (Java).

Листинг программ должен быть оформлен на листах формата А4 и в электронном виде.

Листинг в обязательном порядке должен содержать комментарии.

Номер варианта для выполнения практических заданий определяется по таблице 2 (прил. 4).

Примечание. Студент может предложить другую предметную область для создания приложения.

План выполнения кейса

1. Знакомство с задачей кейса (обзор кейса, постановка задачи, уточнение задания).

2. Выявление основной проблемы.
3. Рассмотрение альтернативных путей решения.
4. Выбор оптимального решения.
5. Реализация решения.
6. Предоставление отчета.

Постановка задачи

1. Выберите вариант практического задания. Изучите текст заданий. При необходимости выполните уточнение заданий.
2. Выполните анализ предметной области. Выявите проблемы, которые необходимо решить, выполнив задания (разрабатываемого приложения).
3. Выберите инструментальные средства разработки программы. Настройте программную среду для разработки приложений с GUI-интерфейсом.
4. Разработайте алгоритмы для решения задач. Проанализируйте альтернативные варианты реализации алгоритмов.
5. Разработайте приложения с GUI-интерфейсом в выбранной инструментальной среде. Выполните их тестирование и отладку.
6. Предложите вариант реализации приложения с базой данных. Как будет выглядеть схема данных БД?

Основные требования к отчету

Все файлы (листинги программ, скомпилированные файлы и др.) выполненного задания, включая отчет, должны быть размещены в папке, имя которой содержит название дисциплины и фамилию студента. В систему дистанционного обучения должен быть загружен архив этой папки (.zip или 7z.).

Например: ООП_ИвановИИ.zip

В файле – отчете (ООП_ИвановИИ.docx) должны располагаться:

- 1) титульный лист;
- 2) текст заданий и листинги с комментариями;
- 3) скриншоты форм созданного приложения.

Требования к оформлению практических работ представлены в Методических указаниях к содержанию, оформлению и критериям оценивания письменных, практических и лабораторных работ, утвержденных решением Научно-методического совета (протокол № 8 от 07.06.2018 г.).

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

Процедура оценивания знаний, умений, навыков и (или) опыта деятельности обучающихся по дисциплине основана на использовании Положения о балльной и рейтинговой системах оценивания, принятой в институте, и технологической карты дисциплины.

Текущий контроль:

- посещение и работа на лекционных и практических занятиях (собеседование, контрольная работа, круглый стол и дискуссия) - 0-35 баллов;
- письменное задание (реферат) - 0-25 баллов;
- практическое задание (кейс) - 0-50 баллов.

Промежуточная аттестация:

- итоговая работа - 25 баллов.

Максимальное количество баллов по дисциплине – 100.

Максимальное количество баллов по результатам текущего контроля – 75.

Максимальное количество баллов на экзамене – 25.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

В процессе изучения учебной дисциплины «Объектно-ориентированное программирование» следует:

1. Ознакомиться с рабочей программой дисциплины. Рабочая программа содержит перечень разделов и тем, которые необходимо изучить, планы лекционных и практических занятий,

вопросы к текущей и промежуточной аттестации, перечень основной, дополнительной литературы и ресурсов информационно-коммуникационной сети «Интернет» и т.д.

2. Ознакомиться с календарно-тематическим планом самостоятельной работы обучающихся.

3. Посещать теоретические (лекционные), практические занятия, лабораторные работы.

4. При подготовке к практическим занятиям и лабораторным работам, а также при выполнении самостоятельной работы следует использовать методические указания для обучающихся.

Учебный план курса «Объектно-ориентированное программирование» предполагает в основе изучения предмета использовать лекционный материал и основные источники литературы, а в дополнение – методические материалы к практическим занятиям и лабораторным работам.

Кроме традиционных лекций, практических занятий и лабораторных работ (перечень и объем которых указаны) целесообразно в процессе обучения использовать и активные формы обучения.

Примерный перечень активных форм обучения:

- 1) беседы и дискуссии;
- 2) кейсы и практические ситуации;
- 3) индивидуальные творческие задания;
- 4) творческие задания в группах;
- 5) практические работы.

На лекциях студенты должны получить систематизированный материал по теме занятия: основные понятия и положения, классификации изучаемых явлений и информационных процессов, и т.д.

Лабораторные работы и практические занятия предполагают более детальную проработку темы по каждой изучаемой проблеме, анализ теоретических и практических аспектов объектно-ориентированного программирования. Для этого разработаны подробные вопросы, обсуждаемые на занятиях, практические задания, темы рефератов и тесты. При подготовке к практическим занятиям следует акцентировать внимание на значительную часть самостоятельной практической работы студентов.

Для более успешного изучения курса преподавателю следует постоянно отсылать студентов к учебникам, периодической печати. Освоение всех разделов курса предполагает приобретение студентами умений самостоятельного анализа языков, средств и механизмов объектно-ориентированного программирования умение работать с научной литературой.

При изучении курса наряду с овладением студентами теоретическими положениями курса уделяется внимание приобретению практических умений с тем, чтобы они смогли успешно применять их в своей профессиональной деятельности.

Большое значение при проверке знаний и умений придается тестированию и подготовке рефератов по темам курса.

Активные формы проведения занятий открывают большие возможности для проверки усвоения теоретического и практического материала.

Основная учебная литература, представленная учебниками и учебными пособиями, охватывает все разделы программы по дисциплине «Объектно-ориентированное программирование». Она изучается студентами в процессе подготовки к практическим занятиям, лабораторным работам, зачету, экзамену. Дополнительная учебная литература рекомендуется для самостоятельной работы по подготовке к практическим занятиям и лабораторным работам, при написании рефератов.

Описание возможностей изучения дисциплины лицами с ОВЗ и инвалидами

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в

печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями опорнодвигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

При проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями. Эти средства могут быть предоставлены вузом или могут использоваться собственные технические средства. Проведение процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья допускается с использованием дистанционных образовательных технологий.

При необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на выполнение заданий текущего контроля. Процедура проведения промежуточной аттестации для обучающихся с ограниченными возможностями здоровья и обучающихся инвалидов устанавливается с учётом индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

Технологическая карта дисциплины

Наименование дисциплины	Объектно-ориентированное программирование
Количество зачетных единиц	4
Форма промежуточной аттестации	Зачет

№	Виды учебной деятельности студентов	Форма отчетности	Баллы (максимум)
Текущий контроль			
1	Посещение и работа на лекционных и практических занятиях (собеседование, контрольная работа, круглый стол и		
2	Выполнение письменного задания (реферат)	Письменная работа	
3	Выполнение практического задания (кейс)	Письменная работа	
Промежуточная аттестация			
4	Выполнение итоговой работы	Итоговая работа, тест	
Итого по дисциплине:			100

« ____ » _____ 20__ г.

Преподаватель

(уч. степень, уч. звание, должность, ФИО преподавателя)

/

Подпись

Технологическая карта дисциплины

Наименование дисциплины	Объектно-ориентированное программирование
Количество зачетных единиц	3
Форма промежуточной аттестации	Экзамен

№	Виды учебной деятельности студентов	Форма отчетности	Баллы (максимум)
Текущий контроль			
1	Посещение и работа на лекционных и практических занятиях (собеседование, контрольная работа, круглый стол и		
2	Выполнение письменного задания (реферат)	Письменная работа	
3	Выполнение практического задания (кейс)	Письменная работа	
Промежуточная аттестация			
4	Выполнение итоговой работы	Итоговая работа, тест	
Итого по дисциплине:			100

« ____ » _____ 20__ г.

Преподаватель

(уч. степень, уч. звание, должность, ФИО преподавателя)

/

Подпись

Тематическое планирование самостоятельной работы студентов

Тема, раздел	Очная форма	Заочная форма	Очно-заочная форма	Задания для самостоятельной работы	Форма контроля
1. Основы объектно-ориентированного программирования. Введение в технологию JAVA	6	22	18	- изучение проблем, не выносимых на лекции; - классификация языков ООП; - подготовка к практическим занятиям; - подготовка к лабораторным работам; - подготовка тематических рефератов и презентаций.	- дополненный конспект; - практическое задание; - отчет по лабораторной работе; - реферат; - тест.
2. Лексика языка Java	12	24	18	- изучение проблем, не выносимых на лекции; - подготовка к практическим занятиям; - подготовка к лабораторным работам; - подготовка тематических рефератов и презентаций.	- дополненный конспект; - практическое задание; - отчет по лабораторной работе; - реферат; - тест.
3. Типы данных	12	24	18	- изучение проблем, не выносимых на лекции; - подготовка к практическим занятиям; - подготовка к лабораторным работам; - подготовка тематических рефератов и презентаций.	- дополненный конспект; - практическое задание; - отчет по лабораторной работе; - реферат; - тест.
4. Code conventions for the JAVA (CCJ)	15	28	18	- изучение оригинала CCJ; документа - изучение проблем, не выносимых на лекции; - подготовка к практическим занятиям; - подготовка к лабораторным работам; - подготовка тематических рефератов и презентаций.	- дополненный конспект; - практическое задание; - отчет по лабораторной работе; - реферат; - тест.

5. Классы, библиотеки, интерфейсы, пакеты	16	30	16	<ul style="list-style-type: none"> - составление таблицы «Библиотеки: описание»; - изучение проблем, не выносимых на лекции; - подготовка к практическим занятиям; - подготовка к лабораторным работам; - подготовка тематических рефератов и презентаций. 	<ul style="list-style-type: none"> - дополненный конспект; - практическое задание; - отчет по лабораторной работе; - реферат; - тест.
6. Классы, библиотеки, интерфейсы, пакеты	8	20	15	<ul style="list-style-type: none"> - изучение проблем, не выносимых на лекции; - подготовка к практическим занятиям; - подготовка к лабораторным работам; - подготовка тематических рефератов и презентаций. 	<ul style="list-style-type: none"> - дополненный конспект; - практическое задание; - отчет по лабораторной работе; - реферат; - тест.
7. Java API	9	25	15	<ul style="list-style-type: none"> - изучение проблем, не выносимых на лекции; - подготовка к практическим занятиям; - подготовка к лабораторным работам; - подготовка тематических рефератов и презентаций. 	<ul style="list-style-type: none"> - дополненный конспект; - практическое задание; - отчет по лабораторной работе; - реферат; - тест.
8. Дженерики и коллекции	6	14	15	<ul style="list-style-type: none"> - изучение проблем, не выносимых на лекции; - подготовка к практическим занятиям; - подготовка к лабораторным работам; - подготовка тематических рефератов и презентаций. 	<ul style="list-style-type: none"> - дополненный конспект; - практическое задание; - отчет по лабораторной работе; - реферат; - тест.
9. Разработка приложения с GUI-интерфейсом		32	16	<ul style="list-style-type: none"> - изучение проблем, не выносимых на лекции; - подготовка к практическим занятиям; - подготовка к лабораторным работам; - подготовка тематических рефератов и презентаций. 	<ul style="list-style-type: none"> - дополненный конспект; - практическое задание; - отчет по лабораторной работе; - реферат; - тест.
ИТОГО	84	219	149		

Показатели и критерии оценивания компетенций на этапе текущего контроля

№ п/п	Показатели оценивания	Критерии оценивания	Шкала оценивания
1	Посещение и работа на лекционных и практических занятиях (собеседование, контрольная работа, круглый стол и дискуссия)	<p>1. Посещение занятий: а) посещение лекционных и практических занятий, б) соблюдение дисциплины.</p> <p>2. Работа на лекционных занятиях: а) ведение конспекта лекций, б) уровень освоения теоретического материала, в) активность на лекции, умение формулировать вопросы лектору.</p> <p>3. Работа на практических занятиях: а) уровень знания учебно-программного материала, б) умение выполнять задания, предусмотренные программой курса, в) практические навыки работы с освоенным материалом.</p>	0-35
2	Письменное задание	<p>1. Новизна текста: а) актуальность темы исследования; б) новизна и самостоятельность в постановке проблемы, формулирование нового аспекта известной проблемы в установлении новых связей (межпредметных, внутрипредметных, интеграционных); в) умение работать с исследованиями, критической литературой, систематизировать и структурировать материал; г) явленность авторской позиции, самостоятельность оценок и суждений; д) стилевое единство текста, единство жанровых черт.</p> <p>2. Степень раскрытия сущности вопроса: а) соответствие плана теме письменного задания; б) соответствие содержания теме и плану письменного задания; в) полнота и глубина знаний по теме; г) обоснованность способов и методов работы с материалом; д) умение обобщать, делать выводы, сопоставлять различные точки зрения по одному вопросу (проблеме).</p> <p>3. Обоснованность выбора источников: а) оценка использованной литературы: привлечены ли наиболее известные работы по теме исследования (в т.ч. журнальные публикации последних лет, последние статистические данные, сводки, справки и т.д.).</p>	0-25

		4. Соблюдение требований к оформлению: а) насколько верно оформлены ссылки на используемую литературу, список литературы; б) оценка грамотности и культуры изложения (в т.ч. орфографической, пунктуационной, стилистической культуры), владение терминологией; в) соблюдение требований к объёму письменного задания.	
3	Практическое задание	<p>1. Анализ проблемы: а) умение верно, комплексно и в соответствии с действительностью выделить причины возникновения проблемы, описанной в практическом задании.</p> <p>2. Структурирование проблем: а) насколько четко, логично, последовательно были изложены проблемы, участники проблемы, последствия проблемы, риски для объекта.</p> <p>3. Предложение стратегических альтернатив: а) количество вариантов решения проблемы, б) умение связать теорию с практикой при решении проблем.</p> <p>4. Обоснование решения: а) насколько аргументирована позиция относительно предложенного решения практического задания; б) уровень владения профессиональной терминологией.</p> <p>5. Логичность изложения материала: а) насколько соблюдены общепринятые нормы логики в предложенном решении, б) насколько предложенный план может быть реализован в текущих условиях.</p>	0-50

Показатели и критерии оценивания компетенций на этапе промежуточной аттестации

№ п/п	Показатели оценивания	Критерии оценивания	Шкала оценивания
1	Итоговая работа	Количество баллов за тест пропорционально количеству правильных ответов на тестовые задания. После прохождения теста суммируются результаты выполнения всех заданий для выставления общей оценки за тест.	0-25

Номер темы для выполнения реферата

Буква фамилии	а	б	в	г	д	е	ж	з	и	к	л	м	н	о
Номер темы реферата	1 или 15	2 или 16	3 или 17	4 или 18	5 или 19	6 или 20	7 или 14	8 или 13	9 или 12	10 или 1	11 или 2	12 или 3	13 или 4	14 или 5
Буква фамилии	п	р	с	т	у	ф	х	ц	ч	ш	щ	э	ю	я
Номер темы реферата	15 или 6	16 или 7	17 или 8	18 или 9	19 или 10	20 или 4	1 или 5	2 или 6	3 или 7	4 или 8	5 или 7	6 или 8	7 или 9	8 или 10

Примерная тематика рефератов

1 часть

1. Объектно-ориентированный подход к программированию.
2. Объектно-ориентированное программирование и программная инженерия.
3. Объектно-ориентированные языки программирования.
4. Решение прикладных задач различных классов с помощью Java-приложений.
5. Программные средства (CASE-средства) объектно-ориентированного подхода проектирования.
6. Автоматизация бизнес-процессов с помощью Java-приложений.
7. Интегрированные среды разработки приложений на объектно-ориентированном языке.
8. Этапы разработки программы на объектно-ориентированном языке. Этап определения цели и задач, разработки плана.
9. Значение ССJ для программистов.
10. Application Programming Interface – интерфейс прикладного программирования.
11. Принципы объектно-ориентированного дизайна классов.
12. Полиморфизм, абстрактные классы и интерфейсы.
13. Язык программирования Python: лексика, синтаксис, основные конструкции, примеры программ.
14. Технология Microsoft .NET.
15. Язык программирования C++: лексика, синтаксис, основные конструкции, примеры программ.
16. Язык программирования C#: лексика, синтаксис, основные конструкции, примеры программ.
17. Язык программирования Objective-C: практические приемы разработки приложений.
18. Язык программирования Ruby: лексика, синтаксис, основные конструкции, примеры программ.
19. Язык программирования Swift: практические приемы разработки приложений.
20. Инструментальные средства разработки программ. Принципы выбора оптимальных языков и инструментов.

2 часть

1. Коллекции в Java.
2. Потоки ввода-вывода.
3. Работа с файлами.
4. Работа со строками.

5. Встроенные функциональные интерфейсы.
6. Лямбда-выражения.
7. Многопоточное программирование.
8. Математические вычисления и класс Math.
9. Большие числа BigInteger и BigDecimal.
10. Работа с датами. LocalDate.
11. Работа с текстом и многоязыковая поддержка.
12. Использование легковесных процессов.
13. Основы сетевого взаимодействия.
14. Работа с потоками ввода-вывода.
15. Сериализация объектов.
16. RMI – вызов удаленных методов.
17. Графический интерфейс пользователя в приложениях Java.
18. Базы данных и Java-приложения.
19. Технология Java-JDBC (концепция и применение).
20. Работа с XML.

1 часть

Практическое задание на тему «Объектно-ориентированное программирование: создание консольных программ»

Таблица 1

Первая буква фамилии студента	№ варианта	№ практических заданий
А, Б	1.	1, 9, 11, 15, 21, 25
В, Г	2.	2, 9, 12, 16, 22, 26
Д, Е, Ё	3.	3, 10, 13, 17, 23, 27
Ж, З, И	4.	4, 9, 14, 18, 24, 28
К, Л, М	5.	5, 10, 11, 19, 21, 29
Н, О, П	6.	6, 10, 14, 20, 22, 25
Р, С, Т	7.	7, 9, 14, 15, 23, 26
У, Ф, Х	8.	1, 10, 13, 16, 24, 27
Ц, Ч, Ш, Щ	9.	2, 9, 12, 19, 22, 28
Э, Ю, Я	10.	8, 9, 13, 20, 21, 29

Практические задания

Задание 1. Вычислите булевы выражения для $a = \text{true}$, $b = \text{false}$, $c = \text{false}$:

$$(a \mid !c) \parallel !(a == b);$$

$$(a \wedge !b) \& (a \wedge !\text{false});$$

Задание 2. Вычислите булевы выражения для $a = \text{true}$, $b = \text{false}$, $c = \text{false}$:

$$(b == !a) \& (a \parallel !b);$$

$$!((a \&\& b) \& (c == !\text{true}) \wedge (a \mid b));$$

Задание 3. Вычислите булевы выражения для $a = \text{true}$, $b = \text{false}$, $c = \text{false}$:

$$(a \wedge !b) \& (a \wedge !\text{false});$$

$$(b == !a) \& (a \parallel !b);$$

Задание 4. Инициализируйте переменные a , b , c , d присвойте им целочисленные значения (типа `int`) и выполните вычисление выражений, используя арифметические операторы, тернарный оператор (проверяя делитель):

$$\underline{a}/(b - c)$$

$$\underline{(a+b)}/\underline{(2a-c)}$$

Начальные значения: $a=3$; $b=2$; $c=4$; $d=1$.

Задание 5. Инициализируйте переменные a , b , c , d присвойте им целочисленные значения (типа `int`) и выполните вычисление выражений, используя арифметические операторы, тернарный оператор (проверяя делитель):

$$\underline{(a+b)}/\underline{(2a-c)}$$

$$\underline{(a\%b)}/\underline{(d-1)}$$

Начальные значения: $a=3$; $b=2$; $c=4$; $d=1$.

Задание 6. Объявите переменные a , b присвойте им целочисленные значения (типа `int`) и выполните вычисление выражений, используя битовые операторы:

$c = a | b;$

$d = a \& b;$

Начальные значения: $a=13; b = 14.$

Задание 7. Объявите переменные a, b присвойте им целочисленные значения (типа `int`) и выполните вычисление выражений, используя битовые операторы:

$e = a \wedge c;$

$f = \sim e \& 0x0f$

Начальные значения: $a=13; b = 14.$

Задание 8. Вычислить выражение $A * (B/3,14) + (C * 3)$ для действительных чисел A, B, C . Значение 3,14 определить в `final`-переменной. Значения A, B – это "минимальные" дробные литералы, а C – литерал, представленный с помощью порядка.

Задание 9. Выполните вычисление выражений для действительных чисел a, b, c, d (с проверкой неравенства нулю делителя). Ответы вывести в двух вариантах: полученные действительные значения и округленные до ближайшего целого.

$(2 * (a - 3) / (a - b * c))$

$(a - b \% c) / (3 * d + c)$

Задание 10. Общий заработок бригады рабочих составил n руб (значение n вводится с клавиатуры). Его нужно разделить между 5 рабочими. При этом учитывая, что процент выполненных работ каждого рабочего от общего объема работ (100%) разный (например, 1 рабочий выполнил 18% работ, второй – 17% и т.д.). Процент выполненных работ для каждого рабочего водится с клавиатуры. Найдите заработок каждого рабочего.

Задание 11. Создайте массив `arr1`, состоящий из n целых чисел. Размерность массива n и значения элементов массива должны вводиться с клавиатуры. При вводе должна выполняться проверка возможности считать с потока ввода `int`. Получите массив `arr2`, значения элементов которого равны квадратам соответствующих значений массива `arr1`. Выведете элементы массивов на консоль в одну строку.

Задание 12. Создайте массив `arr`, содержащий 10 строк (значения вводятся с клавиатуры). Выполните сортировку массива `arr`. Из массива `arr` получите два массива: `arr1`, `arr2`. Массив `arr1` должен содержать только те значения, которые в массиве `arr` соответствовали элементам с нечетными индексами. Массив `arr2` должен содержать только те значения, которые в массиве `arr` соответствовали элементам с четными индексами. Массивы `arr1`, `arr2` проверьте на идентичность.

Задание 13. Создайте массив, состоящий из n целых чисел. Размерность массива n и значения элементов массива должны вводиться с клавиатуры. Выведите значения из массива. Ввод и вывод массива организовать в цикле.

Задание 14. Создайте массив, состоящий из n целых чисел. Размерность массива n должна вводиться с клавиатуры. Значения элементов должны быть равны удвоенным значениям соответствующих индексов (например, значение элемента с индексом 3 равно 6). Выведите значения из массива. Заполнение и вывод массива организовать в цикле. Найти сумму значений полученного массива.

Задание 15. В магазине по продаже дисков действует следующая система скидок:

- при покупке от 5 до 10 дисков скидка 5%
- при покупке от 11 до 15 дисков скидка 8%
- при покупке более 15 дисков скидка 10%

Стоимость одного диска составляет n руб. Напишите программу, которая рассчитывает общую стоимость покупки в зависимости от количества приобретаемых дисков и скидки.

Задание 16. Начальный вклад в банке составил A рублей. Через сколько лет он станет больше B рублей. (Каждый год вклад увеличивается на 12%).

Задание 17. В магазине были приобретены следующие товары:

- сахар a кг по цене b руб.
- мука c кг по цене d руб.
- масло f кг по цене g руб.

Создать программу, выполняющую расчет стоимости покупки. Значения a, b, c, d, f, g должны вводиться с клавиатуры. Расчеты выполнить с применением метода (объявление класса с методом, объявление и вызов метода должны быть выполнены в основной программе).

Задание 18. Создать программу, выполняющую расчет отпускных. Расчеты выполнить с применением метода (методов). Все методы должны быть определены в отдельном файле.

Методика расчета заработной платы:

1) Для того чтобы определить размер отпускных, рассчитывают средний размер заработной платы за последние 12 месяцев.

2) Далее исчисляется средний заработок за один рабочий день, для этого полученное число делят на 29,4 — это среднее число дней в месяце. Полученный показатель умножается на количество дней в отпуске.

С клавиатуры должны вводиться следующие значения:

- Заработная плата за каждый из 12 месяцев (использовать цикл);
- Количество дней в отпуске.

Задание 19. Необходимо оклеить обоями стены в комнате. Длина стен равна a м, а высота b м. Рулон обоев имеет длину c м и ширину d м. Сколько будут стоить обои для всей стены, если цена одного рулона k руб. Создать программу, выполняющую расчет стоимости. Значения a, b, c, d, k должны вводиться с клавиатуры.

Задание 20. В пункте обмена валюты были куплены доллары и евро за рубли:

- a долларов по курсу b руб.
- c евро по курсу d руб.

Создать программу, выполняющую расчет стоимости покупки. Значения a, b, c, d должны вводиться с клавиатуры. Расчеты выполнить с применением метода.

В программе *Program3_03.java* должен создаваться экземпляр класса *Kurs* и выполняться вызов необходимых методов. Все методы должны быть определены в файле *Kurs.java*. Переменные a, b, c, d должны быть объявлены в *Kurs.java*.

Дополнительно: найти способ вывода текущей даты.

Задание 21. Создайте абстрактный класс *MathAbstract*. В нем создайте два метода:

- *operMath()* без реализации с параметрами `int a, int b`;
- *show()* без параметров, реализующий вывод строки на экран «Иллюстрация работы с абстрактным классом: арифметические операции».

Создайте неабстрактные классы *MathReal01* – *MathReal05* – производные от *MathAbstract*. В них выполните реализации метода *operMath()*. Реализации соответствуют выполнению основных арифметических операций (сложение, вычитание, умножение, деление, возведение в степень).

В основной программе выполните реализацию: вызов всех методов класса *MathAbstract* и выполнение соответствующих операций. Значения a, b вводятся с клавиатуры.

Задание 22. На рисунке представлено дерево наследования (сверху вниз - от базовых классов к производным) в соответствии с некоторыми группами товаров.



Продумайте спецификацию классов.

На основе спецификаций создайте **общедоступные классы** (т.е. в них описать структуру данных). Определите **общедоступный конструктор** для каждого класса.

В основной программе выполните реализацию: на основе конструкторов создать по два экземпляра и вывести все данные на экран.

Задание 23. Создать общедоступный класс **Vector**. Поля класса: vx (private int), vy (private int), length (protected double).

- vx, vy – это координаты вектора;
- length – это длина вектора.

Определить общедоступные конструкторы для класса:

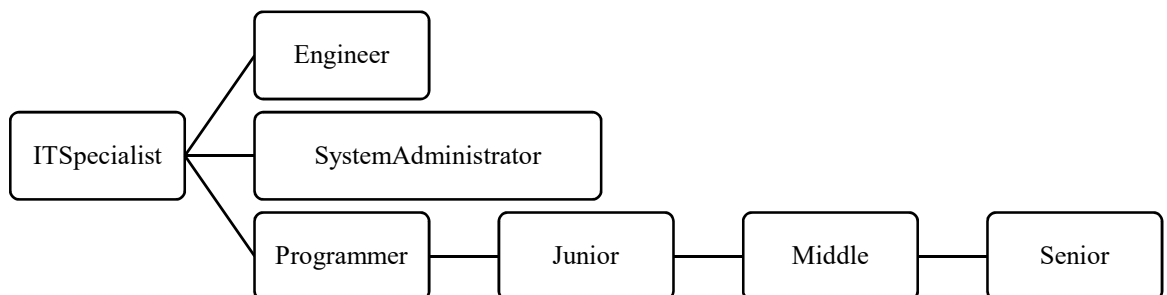
Конструктор 1: два аргумента – **координаты вектора**: int x, int y. Инициализировать поля класса. При этом для инициализации length (длина) использовать формулу вычисления длины вектора по его координатам.

Конструктор 2: четыре аргумента – **координаты точек** – начала и конца вектора: int x1, int y1, int x2, int y2. Инициализировать поля класса. При этом для инициализации length (длина) использовать формулу вычисления длины вектора также по его координатам (т.е. инициализаторы для length в обоих конструкторах одинаковы).

Не использовать this при инициализации переменных и ссылки на конструктор.

В основной программе выполнить реализацию (создать два экземпляра на основе двух разных конструкторов, значения параметров вводятся с клавиатуры) и вывести все данные на экран.

Задание 24. На рисунке представлено дерево наследования (сверху вниз - от базовых классов к производным).

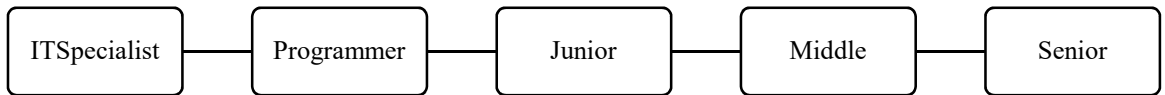


Продумайте спецификацию классов.

На основе спецификаций создайте **общедоступные классы** (т.е. в них описать структуру данных и методы).

В основной программе выполните реализацию: на основе конструкторов создать по одному экземпляру классов: ITSpecialist, Engineer, SystemAdministrator, Programmer, Junior, Middle, Senior.

Выполните приведение типов классов (на основе «ветки»):



Задание 25. Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать критерий (критерии) выбора данных и вывести эти данные на консоль.

Спецификация. Student: id, Фамилия, Имя, Отчество, Адрес, Телефон, Специальность, Курс, Группа, Оценки за сессию.

Создать массив объектов. Вывести:

- 1) список студентов заданной специальности
- 2) список студентов, имеющих неудовлетворительные оценки
- 3) список студентов заданной учебной группы
- 4) список студентов, чей средний балл меньше заданного

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**.

Задание 26. Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать критерий (критерии) выбора данных и вывести эти данные на консоль.

Спецификация. Patient: id, Фамилия, Имя, Отчество, Адрес, Телефон, Номер медицинской карты, диагноз.

Создать массив объектов. Вывести:

- 1) список пациентов, имеющих данный диагноз;
- 2) список пациентов, номер медицинской карты у которых находится в заданном интервале.

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**.

Задание 27. Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать критерий (критерии) выбора данных и вывести эти данные на консоль.

Спецификация. Book: id, Название, Автор(ы), Издательство, Год издания, Количество страниц, Цена, Переплет.

Создать массив объектов. Вывести:

- 1) список книг заданного автора;
- 2) список книг, выпущенных заданным издательством;
- 3) список книг, выпущенных после заданного года.

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**.

Задание 28. Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать критерий (критерии) выбора данных и вывести эти данные на консоль.

Спецификация. House: id, Номер квартиры, Площадь, Этаж, Количество комнат, Улица, Тип здания, Срок эксплуатации.

Создать массив объектов. Вывести:

- 1) список квартир, имеющих заданное число комнат;
- 2) список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в заданном промежутке;
- 3) список квартир, имеющих площадь, превосходящую заданную.

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**.

Задание 29. Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать критерий (критерии) выбора данных и вывести эти данные на консоль.

Спецификация. Car: id, Марка, Модель, Год выпуска, Цвет, Цена, Регистрационный номер.

Создать массив объектов. Вывести:

- 1) список автомобилей заданной марки;
- 2) список автомобилей заданной модели, которые эксплуатируются больше n лет;
- 3) список автомобилей заданного года выпуска, цена которых больше указанной.

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**.

2 часть
**Практическое задание на тему «Объектно-ориентированное
программирование: создание программ с GUI-интерфейсом»**

Таблица 2

Первая буква фамилии студента	№ варианта	Предметная область
А, Б	1.	Учет расходов (домашняя бухгалтерия) или Анкетирование
В, Г	2.	Учет топливных ресурсов или Тестирование
Д, Е, Ё	3.	Кулинарная книга Или Расписание занятий или режим дня
Ж, З, И	4.	Расписание занятий или режим дня Или Компьютерная игра
К, Л, М	5.	Конвертирование форматов файлов или Анкетирование
Н, О, П	6.	Оболочка для чтения текстовых файлов Или Тестирование
Р, С, Т	7.	Анкетирование Или Расписание занятий или режим дня
У, Ф, Х	8.	Тестирование Или Расписание занятий или режим дня
Ц, Ч, Ш, Щ	9.	Компьютерная игра Или Оболочка для чтения текстовых файлов
Э, Ю, Я	10.	Электронное учебное пособие Или Тестирование

Тестовые задания по дисциплине «Объектно-ориентированное программирование»

1 часть

1. Какое утверждения относительно языка Java является верным

- 1) язык Java является объектно-ориентированным
- 2) язык Java является расширением HTML
- 3) язык Java является языком структурного программирования
- 4) язык JavaScript – это версия языка Java

2. Компанией-разработчиком языка Java является:

- 1) Microsoft Corporation
- 2) Sun Microsystems
- 3) IBM
- 4) SAP SE
- 5) Dassault Systemes

3. JRE – это

- 1) кодовое название проекта для разработки языка Java
- 2) минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений
- 3) комплект разработчика приложений на языке Java
- 4) набор сгруппированных в пакеты классов для работы с языком Java

4. JDK – это:

- 1) минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений
- 2) комплект разработчика приложений на языке Java
- 3) первое демонстрационное устройство на основе платформы Java
- 4) инструментальная среда разработки Java-приложений

5. В составе JRE отсутствуют (предусмотрено несколько вариантов ответа)

- 1) компилятор Java
- 2) библиотеки классов Java
- 3) виртуальная машина
- 4) инструментальные средства разработки Java-приложений

6. Стандартной версией платформы Java 2 является

- 1) Extended Edition
- 2) Java Card
- 3) Micro Edition
- 4) Enterprise Edition
- 5) Standard Edition

7. Интерактивное приложение для веб-страниц называется

- 1) Апплетом
- 2) Мидлетом
- 3) Сервлетом

8. Какое из следующих утверждений, касающихся языка Java, является неверным

- 1) Компилятор Java генерирует машинные команды для процессора ЭВМ
- 2) Компилятор Java создаёт промежуточный код – байт-код для виртуальной машины Java
- 3) В основе JVM лежит интерпретатор байт-кода Java)
- 4) По сути JVM – это виртуальный компьютер, размещенный в оперативной памяти реальной машины

9. Java API – это

- 1) минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений
- 2) комплект разработчика приложений на языке Java
- 3) набор сгруппированных в пакеты классов для работы с языком Java
- 4) инструментальная среда разработки Java-приложений

10. Какое из следующих утверждений, касающихся языка Java, является неверным

- 1) Программа, выполняемая в режиме интерпретации работает существенно быстрее, чем программа, полученная в виде машинного кода после компиляции
- 2) С целью повышения производительности в платформу Java 2 внесены дополнения, в частности Just-In-Time
- 3) С целью повышения производительности в платформу Java 2 внесены дополнения, в частности Java HotSpot
- 4) Компиляторы JIT обеспечивают компиляцию байт-кода в машинный код в реальном времени при выполнении программы («на лету»)
- 5) Виртуальная машина Java HotSpot преобразовывает в машинный код только те фрагменты байт-кода, которые влияют на производительность

11. Верификация байт-кода означает следующее:

- 1) Некорректный байткод будет отвергнут перед исполнением
- 2) Байткод преобразуется в машинный код
- 3) Увеличение производительности программных систем, использующих байт-код, путём компиляции байт-кода в машинный код в реальном времени
- 4) Оптимизация кода во время его выполнения

12. Какое из следующих утверждений, касающихся языка Java, является неверным:

- 1) Сборка мусора – одна из форм автоматического управления памятью
- 2) Сборщик мусора периодически освобождает память, удаляя объекты, которые уже не будут востребованы приложениями
- 3) В JAVA требуется ручное освобождение памяти
- 4) В JAVA нет арифметики указателей

13. Какое из следующих утверждений, касающихся языка Java, является верным:

- 1) Указатель – это final-переменная, значение которой задается программистом при написании программы.
- 2) Указатель – это переменная, диапазон значений которой состоит из адресов ячеек памяти или специального значения – нулевого адреса
- 3) Указатель – это отметка, создаваемая во время отладки программы и указывающая на ошибку в программном коде
- 4) Указатель – это тип переменной, задаваемый во время ее объявления

14. Технологией, предназначенной для создания графических интерфейсов корпоративных приложений и бизнеса является:

- 1) Java Enterprise Edition
- 2) Java Micro Edition
- 3) JavaFX
- 4) Java Card

15. Основным отличием JDK от JRE является:

- 1) Наличие среды исполнения – компилятора
- 2) Наличие стандартных библиотек классов Java
- 3) Бесплатное распространение

16. Правильным порядком выполнения шагов установки Java и инструментария является:

- 1) установка инструментальной среды, установка пакета JDK, настройка переменных сред
- 2) настройка переменных сред, установка инструментальной среды, установка пакета JDK

- 3) установка инструментальной среды, настройка переменных сред, установка пакета JDK,
- 4) установка пакета JDK, настройка переменных сред, установка инструментальной среды

17. Расширением файла в языке java, содержащим байткод, является:

- 1) .java
- 2) .byte
- 3) .javac
- 4) .class

18. Расширением файла, содержащим исходный код программы на языке java, является:

- 1) .java
- 2) .jre
- 3) .jdk
- 4) .class

19. При успешной компиляции программы на языке java создается файл с расширением:

- 1) .java
- 2) .exe
- 3) .javac
- 4) .class

20. Выберите верное утверждение относительно программы HelloWorld.java:

```
class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

- 1) Public – это основной метод, main() – это ничего не возвращающий метод, void – это модификатор доступа
- 2) Public – это модификатор доступа, main() – это основной метод, void – это модификатор доступа
- 3) Public позволяет методу main() вызываться без потребности создавать дубликат класса, main() – это идентификатор класса, void – это ничего не возвращающий метод
- 4) Public – это модификатор доступа, main() – это основной метод, void – это ничего не возвращающий метод

21. Исходный код программы Code.java необходимо скомпилировать и затем запустить файл на выполнение (в командной строке). Правильными командами для этих действий являются

- 1) javac Code.java, java Code.java
- 2) javac Code.java, java Code.class
- 3) javac Code.java, java Code
- 4) javac Code.class, java Code.java
- 5) javac Code.class, java Code

22. Стандартным обозначением конца строки в Java-программах является:

- 1) \n
- 2) \e
- 3) \k
- 4) \r

23. Стандартным обозначением возврата каретки в Java-программах является:

- 1) \n
- 2) \f
- 3) \b
- 4) \r

24. Java является строго типизированным языком. Это означает, что

- 1) все типы данных разделены на классификационные группы

- 2) во время компиляции должен быть известен тип каждой переменной и каждого выражения
- 3) не может быть выполнено преобразование одного типа данным к другому
- 4) для каждого типа данных определен свой набор, строго допустимых операций

25. Ссылочными типами данных в языке Java являются (предусмотрено несколько вариантов ответа)

- 1) массивы
- 2) булев тип
- 3) классы
- 4) символьный тип
- 5) интерфейсы

26. Из следующих утверждений, касающихся языка java, выберите одно неверное

- 1) для работы с потоком ввода необходимо создать объект класса Scanner
- 2) стандартный поток ввода (клавиатура) в Java представлен объектом System.out
- 3) для ввода данных используется класс Scanner, который надо импортировать
- 4) в классе Scanner есть методы для чтения очередного символа заданного типа со стандартного потока ввода, а также для проверки существования такого символа

27. В структуре класса на языке java порядок элементов должен быть следующим

- 1) заголовок класса, документирующий комментарий, статические поля, нестатические поля, конструкторы, методы
- 2) статические поля, документирующий комментарий, заголовок класса, нестатические поля, конструкторы, методы
- 3) документирующий комментарий, заголовок класса, статические поля, нестатические поля, методы, конструкторы
- 4) документирующий комментарий, заголовок класса, статические поля, нестатические поля, конструкторы, методы

28. Первое действие, которое происходит при входе в конструкцию if –else:

- 1) выполняются инструкции блока if
- 2) выполняются инструкции блока else
- 3) выполняется проверка условия
- 4) все перечисленные варианты выполняются одновременно

29. Из следующих утверждений выберите одно неверное:

- 1) конструкция switch позволяет передавать управление блоку кода, обозначенному константой целого типа
- 2) при отсутствии команды break в каждом блоке case конструкция switch не будет работать
- 3) в конструкции switch блок default может отсутствовать

30. Программный блок, в котором генерируется исключение – это

- 1) try
- 2) catch
- 3) finally

31. Какие два метода разработки называют классическими?

- 1) архитектурный и нисходящий
- 2) восходящий и архитектурный
- 3) архитектурный и конструктивный
- 4) нисходящий и восходящий

32. Соотнесите название видов контроля с их определениями

- 1) Статический контроль
- 2) Смежный контроль сверху
- 3) Сквозной контроль
- 4) Смежный контроль снизу

а) - это контроль со стороны разработчиков архитектуры и внешнего описания ПС.

б) - это мысленное прокручивание (проверка) структуры программы при выполнении заранее разработанных тестов.

в)- это оценка структуры программы, насколько хорошо программа разбита на модули с учетом значений рассмотренных выше основных характеристик модуля.

г) - это контроль спецификации модулей со стороны разработчиков этих модулей.

33. Укажите правильную последовательность разработки программного модуля

- 1) Изучение и проверка спецификации модуля, выбор языка программирования.
- 2) Компиляция модуля.
- 3) Проверка модуля.
- 4) Шлифовка текста модуля.
- 5) Выбор алгоритма и структуры данных.
- 6) Программирование (кодирование) модуля.

34. Специальные символы, которые сообщают транслятору о том, что необходимо выполнить операцию с некоторыми операндами – это

- 1) Идентификаторы
- 2) Разделители
- 3) Операторы
- 4) Литералы

35. Позволяют задать в программе значения для числовых, символьных и строковых выражений

- 1) Идентификаторы
- 2) Разделители
- 3) Операторы
- 4) Литералы

36. Какому термину объектно-ориентированного программирования соответствует определение: «...» - это шаблон, на котором основаны похожие объекты

- 1) метод
- 2) поле
- 3) класс
- 4) пакет

37. Имена, которые даются различным элементам языка для упрощения доступа к ним, называются

- 1) Идентификаторами
- 2) Разделителями
- 3) Операторами
- 4) Переменными

38. К какому из свойств объектно-ориентированного программирования относят следующее утверждение: «Возможность иметь разные тела у методов с одним и тем же именем, но принадлежащих различным классам в одной иерархии наследования»

- 1) наследование;
- 2) полиморфизм;
- 3) инкапсуляция;
- 4) множественное наследование.

39. Выберите правильное объяснение понятию «интеграционное тестирование»

- 1) состоит в тестировании методов, унаследованных классом от своего базового класса.
- 2) представляет собой тестирование того, как отдельные элементы программы работают вместе.
- 3) проверяет всю программную систему целиком и строится в большинстве случаев по принципу «черного ящика».

40. Какому термину соответствует определение «...» – это деятельность, направленная на обеспечение необходимых условий для работы коллектива разработчиков ПС, на планирование и контроль деятельности этого коллектива с целью обеспечения требуемого качества ПС, выполнения сроков и бюджета разработки ПС

- 1) Управление разработкой ПС
- 2) Тестирование ПС
- 3) Отладка ПС
- 4) Эксплуатация ПС

41. Соотнесите названия документов, создаваемых при проектировании программных средств, с их описанием

- 1) Планы, оценки, расписания
- 2) Заметки и переписка
- 3) Стандарты
- 4) Рабочие документы
- 5) Отчеты об использовании ресурсов в процессе разработки

а) Это основные технические документы, обеспечивающие связь между разработчиками.

б) Эти документы предписывают разработчикам, каким принципам, правилам, соглашениям они должны следовать в процессе разработки ПС.

в) Эти документы создаются менеджерами для прогнозирования и управления процессами разработки и сопровождения ПС.

г) Эти документы фиксируют различные детали взаимодействия между менеджерами и разработчиками.

д) Создаются менеджерами.

42. Какое из следующих утверждений является верным

- 1) Указатель – это final-переменная, значение которой задается программистом при написании программы.
- 2) Указатель – это переменная, диапазон значений которой состоит из адресов ячеек памяти или специального значения – нулевого адреса
- 3) Указатель – это отметка, создаваемая во время отладки программы и указывающая на ошибку в программном коде
- 4) Указатель – это тип переменной, задаваемый во время ее объявления

43. Автоматическое переименование переменных, классов является примером

- 1) Автодополнения
- 2) Рефакторинга
- 3) Интерактивной отладки

44. Ссылочными типами данных являются (правильных ответов несколько)

- 1) массивы
- 2) булев тип
- 3) классы
- 4) символьный тип
- 5) интерфейсы
- 6) строки

45. Запись алгоритма на языке программирования называется

- 1) отладкой
- 2) тестированием
- 3) кодированием
- 4) проектированием

46. Из следующих утверждений выберите одно неверное

- 1) в языке Java поддерживается только простое наследование

- 2) в языке Java поддерживается не только простое, но и множественное наследование
- 3) отсутствие множественного наследования в Java компенсируется использованием интерфейсов
- 47. Если для класса не указан его суперкласс, по умолчанию этот класс расширяется классом**
- 1) java.lang.Super
 - 2) java.lang.Main
 - 3) java.lang.Object
 - 4) java.lang.SuperClass
- 48. Объявление класса в Java состоит из заголовка и тела. В общем виде может быть представлено следующим образом**
- 1) [модификаторы] class Имя_класса [implements Имя(Список)_интерфейса(ов)] [extends Имя_суперкласса] {}
 - 2) [модификаторы] class Имя_класса [extends Имя_суперкласса] [implements Имя(Список)_интерфейса(ов)] {}
 - 3) [модификаторы] [implements Имя(Список)_интерфейса(ов)] class Имя_класса [extends Имя_суперкласса] {}
 - 4) [implements Имя(Список)_интерфейса(ов)] [модификаторы] class Имя_класса [extends Имя_суперкласса] {}
- 49. Из следующих утверждений выберите одно верное**
- 1) Производный класс наследует структуру данных и может добавлять свою собственную структуру, поведение своего суперкласса не наследуется
 - 2) Производный класс наследует структуру данных и поведение своего суперкласса и может добавлять свои собственные структуру данных и поведение
 - 3) Производный класс наследует структуру данных и поведение своего суперкласса, но не может добавлять свои собственные структуру данных и поведение
 - 4) Производный класс наследует поведение своего суперкласса, структура данных своего суперкласса не наследуется
- 50. К элементам класса относятся (предусмотрено несколько вариантов ответа)**
- 1) поля
 - 2) блоки инициализации
 - 3) методы
 - 4) внутренние классы
 - 5) конструкторы

2 часть

- 1. Спецификаторами доступа класса являются (предусмотрено несколько вариантов ответа)**
- 1) public
 - 2) abstract
 - 3) final
 - 4) default
- 2. Не допускается создание наследников класса с модификатором**
- 1) public
 - 2) abstract
 - 3) final
 - 4) default
- 3. Не имеют имена и не передаются по наследству (предусмотрено несколько вариантов ответа)**
- 1) конструкторы
 - 2) методы

3) инициализаторы

4) поля

5) внутренние классы

4. Полями класса являются

1) параметры методов, конструкторов, обработчиков исключений

2) переменные конструктора или блока инициализации

3) переменные класса, которые объявляются вне методов и доступны всем методам класса

4) переменные блока операторов {...}

5. Не допускается создание экземпляров класса с модификатором

1) public

2) abstract

3) final

4) default

6. Модификаторами доступа полей класса являются (предусмотрено несколько вариантов ответа)

1) private

2) transient

3) static

4) public

5) final

6) default

7) protected

8) volatile

7. Сигнатурой метода называется

1) комбинация типа возвращаемого значения и имени-идентификатора метода

2) комбинация типа возвращаемого значения, имени-идентификатора метода и списка типов параметров

3) комбинация модификатора доступа, имени-идентификатора метода и списка типов параметров

4) комбинация имени-идентификатора метода и списка типов параметров

8. Модификаторами доступа методов класса являются (предусмотрено несколько вариантов ответа)

1) abstract

2) private

3) static

4) public

5) final

6) default

7) protected

8) native

9. Из следующих утверждений выберите одно неверное

1) Выражений return в теле одного метода может быть несколько

2) Если в заголовке метода указан тип возвращаемого значения, то в теле метода обязательно должно встречаться return-выражение

3) Поля и методы не могут иметь одинаковые имена

4) В методе без возвращаемого значения (void) также можно использовать выражение return без каких либо аргументов

10. Необходимо получить случайное целое число из [2;5]. Правильным значением выражения с функцией Random является

1) ((int)(Math.random()*5+2))

2) ((int)(Math.random()*4+2))

3) ((int)(Math.random()*2+5))

4) ((int)(Math.random()*7-2))

11. Перегруженными методами называются методы

- 1) разных классов, но с одинаковыми именами
- 2) одного класса с одинаковыми именами
- 3) одного класса с одинаковыми сигнатурами
- 4) одного класса с одинаковым типом возвращаемого значения

12. Перегрузки для методов реализуют свойство в программировании

- 1) наследование
- 2) полиморфизм
- 3) инкапсуляция

13. Из следующих утверждений выберите неверные (предусмотрено несколько вариантов ответа)

- 1) Класс может содержать несколько конструкторов
- 2) У конструктора не могут быть аргументы (параметры)
- 3) Конструктор вызывается при создании экземпляра класса с помощью оператора new
- 4) Конструктор не возвращает значения
- 5) Класс должен иметь хотя бы один конструктор, определённый явно

14. Сигнатура конструктора определяется (предусмотрено несколько вариантов ответа)

- 1) именем класса
- 2) количеством аргументов
- 3) именами аргументов
- 4) типом аргументов
- 5) порядком следования аргументов
- 6) throws-выражением

15. Модификаторами конструктора могут быть (предусмотрено несколько вариантов ответа)

- 1) abstract
- 2) private
- 3) static
- 4) public
- 5) final
- 6) default
- 7) protected
- 8) native

16. Поля, принадлежащие классу (предусмотрено несколько вариантов ответа)

- 1) статические
- 2) доступны, когда создан объект класса
- 3) доступны без создания объекта класса
- 4) инициализироваться должны до создания объекта класса
- 5) нестатические
- 6) инициализироваться должны после создания объекта класса

17. Инициализаторы

- 1) имеют имя (совпадающее с именем класса), исполняются при создании объектов, не передаются по наследству
- 2) не имеют имен, исполняются при создании объектов и не могут быть вызваны явно, не передаются по наследству
- 3) имеют имя, исполняются при создании объектов и не могут быть вызваны явно, передаются по наследству
- 4) не имеют имен, исполняются при создании объектов и не могут быть вызваны явно, передаются по наследству

18. Из следующих утверждений выберите одно неверное

- 1) абстрактный метод объявляется с ключевым словом `abstract`
- 2) к абстрактному методу можно обращаться, даже в случае, если не описана его реализация
- 3) абстрактный метод не имеет тела
- 4) абстрактный метод не может быть `private`, `native`, `static`

19. Из следующих утверждений выберите неверные (предусмотрено несколько вариантов ответа)

- 1) абстрактный класс можно не объявлять с ключевым словом `abstract` при условии, что в нем есть абстрактный метод
- 2) класс не может быть одновременно `abstract` и `final`
- 3) абстрактный класс должен содержать хотя бы один абстрактный метод
- 4) абстрактный класс может содержать и полностью реализованные методы

20. Из следующих утверждений выберите одно неверное

- 1) интерфейс представляет собой набор методов без реализации (без тел), который впоследствии может быть реализован классом
- 2) класс в Java может реализовывать только один интерфейс, а один интерфейс - использоваться несколькими классами
- 3) для реализации интерфейса класс должен определить полный набор методов, которые объявлены в интерфейсе
- 4) в java есть «готовые» интерфейсы

21. Соккрытие реализации реализуют свойство в программировании

- 1) наследование
- 2) полиморфизм
- 3) инкапсуляция

22. Все методы в интерфейсах по умолчанию считаются

- 1) закрытыми и абстрактными
- 2) открытыми и статическими
- 3) открытыми и абстрактными
- 4) закрытыми и статическими

23. Из следующих утверждений выберите одно неверное

- 1) пакет импортируется в программу с помощью оператора `import`
- 2) собственный пакет можно создать с помощью оператора `new`
- 3) в пределах пакета разрешен неограниченный доступ классов друг к другу
- 4) квалифицированное имя класса включает имя пакета

24. Правильным порядком уровней доступа (от более открытых к менее) является

- 1) `public`, `default`, `protected`, `private`
- 2) `public`, `protected`, `private`, `default`
- 3) `public`, `protected`, `default`, `private`
- 4) `default`, `public`, `protected`, `private`

25. Из следующих утверждений выберите одно верное

- 1) уровни доступа пакетов: `public`, `default`
- 2) пакеты всегда доступны, поэтому у них нет модификаторов доступа
- 3) уровни доступа пакетов: `protected`, `private`
- 4) уровни доступа пакетов – все, кроме `private`

26. Из следующих утверждений выберите одно неверное

- 1) не обязательно указывать размерность при создании объекта `ArrayList`
- 2) после того как массив `ArrayList` создан, он не может расти или уменьшаться
- 3) Элементы `ArrayList` могут быть абсолютно любых типов в том числе и `null`
- 4) `ArrayList` может менять свой размер во время исполнения программы

27. **Уровень доступа, при котором элемент доступен классу, в котором объявлен, наследникам в том же пакете и другим классам в том же пакете, но не доступен классам, в том числе и наследникам, находящимся в других пакетах - это**
- 1) private
 - 2) public
 - 3) default
 - 4) protected
28. **API – это**
- 1) минимальная реализация виртуальной машины, необходимая для исполнения приложений
 - 2) комплект разработчика приложений на языке программирования
 - 3) набор сгруппированных в пакеты классов для работы с языком программирования
 - 4) инструментальная среда разработки приложений
29. **Стандартная библиотека ввода-вывода java - это**
- 1) java.lang
 - 2) java.io
 - 3) java.util
 - 4) java.awt
30. **Обеспечивает основные функциональные возможности языка библиотека**
- 1) java.sql
 - 2) java.lang
 - 3) java.io
 - 4) java.util
31. **Обеспечивает возможности создания графического интерфейса пользователя библиотека**
- 1) java.awt
 - 2) java.applet
 - 3) java.io
 - 4) java.util
32. **Расширяет возможности создания графического интерфейса пользователя библиотека**
- 1) java.beans
 - 2) java.applet
 - 3) java.rmi
 - 4) javax.swing
33. **Поддерживает вызов удалённых методов, т.е. распределенное сетевое программирование библиотека**
- 1) java.beans
 - 2) java.util.jar
 - 3) java.rmi
 - 4) java.awt.event
34. **Поддерживает обработку событий библиотека**
- 1) java.awt.event
 - 2) java.applet
 - 3) java.io
 - 4) java.util.jar
35. **Оконные объекты, в которых можно располагать компоненты, называются**
- 1) Конструкторами
 - 2) Контейнерами
 - 3) Контроллерами
 - 4) Интерфейсами
36. **Panel и Window – это подклассы класса**

- 1) Container
- 2) Canvas
- 3) Frame
- 4) Button
37. **Класс, представляющий собой окно с рамкой и строкой заголовка (с кнопками «Свернуть», «Во весь экран» и «Заккрыть») – это**
 - 1) Window
 - 2) Panel
 - 3) Контроллерами
 - 4) JFrame

38. **В приведенном коде выполняются следующие действия**

```
import java.awt.*;

import java.applet.*;

public class ExampleApplet01 extends Applet {
    public void paint(Graphics g) {
        g.drawString("Пример апплета!", 30, 30);
    }
}
```

- 1) Создается апплет, в котором в позиции экрана (30,30) выводится строка "Пример апплета!"»
- 2) Создается апплет, в котором в позиции экрана (30,30) рисуется точка, а в заголовке окна выводится строка "Пример апплета!"»
- 3) Создается апплет размером 30*30, в котором рисуется точка, а в заголовке окна выводится строка "Пример апплета!"»
- 4) Создается апплет размером 30*30, в котором выводится строка "Пример апплета!"»

39. **В приведенном коде выполняются следующие действия**

```
import java.applet.*;
import java.awt.Graphics;

public class Program651 extends Applet {
    private String date;
    @Override
    public void init() {
        date = new java.util.Date().toString();
    }
    public void paint(Graphics g) {
        g.drawString("Пример апплета!", 30, 30);
        g.drawString(date, 30,100);
    }
}
```

- 1) Создается апплет, в котором в позиции экрана (30,30) выводится строка «Пример апплета!», а в позиции экрана (30,100) системная дата
- 2) Создаются два апплета, в которых в позициях экрана (30,30) и (30,100) соответственно выводятся строки «Пример апплета!» и дата
- 3) Создаются два апплета с размерами (30,30) и (30,100) соответственно, в которых выводятся строки «Пример апплета!» и дата
- 4) Рисуется две точки в позициях (30,30) и (30,100), а в заголовке окна выводятся строки «Пример апплета!» и дата

40. **В приведенном коде допущена ошибка в жизненном цикле апплета. Какие изменения необходимо внести в программный код для устранения ошибки?**

```

import java.awt.*;

import java.applet.*;

public class MyApplet extends Applet {
    public void init() {
    }
    public void start() {
    }
    public void paint(Graphics g) {
        g.drawString("Рисование строки в окне апплета", 10, 50);
    }
    public void destroy() {
    }
    public void stop() {
    }
}

```

- 1) метод paint() вызвать после метода destroy()
- 2) метод destroy() вызвать после метода stop()
- 3) метод init() вызвать после метода destroy()
- 4) метод start() вызвать перед метода init()

41. При вызове методов класса Fxyz (по порядку их следования в классе) и значениях $x=5,5$ и $y=1,5$ будут получены следующие значения

```

class Fxyz {
    static double getSum(double x, double y) {
        return x + y;
    }
    static double getSq(double x, double y) {
        return Math.sqrt(Math.abs(x - y));
    }
    static double getStepen(double x) {
        int p = 2;
        return Math.pow(x, p);
    }
}

```

- 1) 7; 2; 7
- 2) 4; 4; 7
- 3) 7; 4; 30,25
- 4) 7; 2; 30,25

42. В приведенном коде допущена синтаксическая ошибка. Какие изменения необходимо внести в программный код для устранения ошибки?

```

class Point {
    int y = x;
    int x = 3;

    public static void main (String s[]) {
        Point p=new Point();
        System.out.println(p.x+", "+p.y);
    }
}

```

- 1) Использовать ввод данных поля x с консоли
- 2) Изменить тип данных поля y
- 3) Использовать поле x в инициализации поля y после объявления x
- 4) Инициализировать поля после public static void

43. В приведенном коде допущена синтаксическая ошибка. Какие изменения необходимо внести в программный код для устранения ошибки?

```

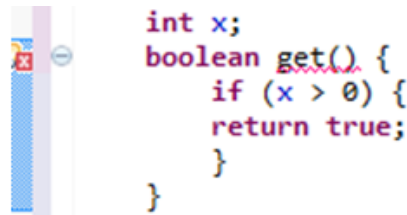
void getXy(double x, final double y) {
    x = Math.cos(x);
    y = Math.sin(x);
}

```

- 1) для аргумента x указать ключевое слово final
 - 2) удалить ключевое слово final аргумента y
 - 3) выражения, вычисляющие x, y записать в одной строке через запятую
 - 4) выражение, вычисляющее y заключить в фигурные скобки
44. Укажите случай, в котором вызов метода getChislo() выбросит исключение типа RuntimeException

```
class Chislo {
    int getChislo(String str) {
        int chislo = Integer.parseInt(str);
        return chislo;
    }
}
```

- 1) если метод getChislo() будет вызван в основной программе более одного раза
 - 2) если строка str будет содержать недопустимые символы
 - 3) если допущена арифметическая ошибка
 - 4) если запрашиваемый метод не существует
45. В приведенном коде допущена ошибка. Укажите, с чем она связана



```
int x;
boolean get() {
    if (x > 0) {
        return true;
    }
}
```

- 1) тело метода содержит return-выражение
 - 2) не при любом развитии событий возвращаемое значение будет сгенерировано
 - 3) имя метода записано со строчного символа
 - 4) тип возвращаемого значения задан неверно
46. При выполнении следующей программы значение переменной sum будет равно

```
public class Program4375 {
    public static void main(String[] args) {
        int sum = 0;
        System.out.println("все нечетные двузначные числа:");
        for (int i = 10; i < 100; i++) {
            if (i % 2 != 0) {
                System.out.print(i + " ");
                sum += i;
            }
        }
        System.out.println("\nсумма = " + sum);
    }
}
```

- 1) 2530
- 2) 2430
- 3) 2475
- 4) 4905

47. Данный этап разработки программы, в частности, включает консультации представителей заказчика по работе с программой и обучение персонала

- 1) публикация
- 2) сопровождение
- 3) отладка
- 4) анализ результатов

48. Тестирование программного обеспечения в целях проверки способности ПО в определённых условиях решать задачи, нужные пользователям, называется

- 1) нагрузочным
- 2) тестированием производительности
- 3) функциональности
- 4) стресс-тестированием

49. Один из видов тестирования программного обеспечения, которое оценивает надёжность и устойчивость системы в условиях превышения пределов нормального функционирования, называется

- 1) нагрузочным
- 2) тестированием производительности
- 3) функциональности
- 4) стресс-тестированием

50. Соотнесите виды тестирования с их целями

- 1) Тестирование при комплексной отладке
- 2) Тестирование архитектуры ПС
- 3) Тестирование внешних функций

а) Применение ПС к конкретным данным, которые могут возникнуть у пользователя, но, возможно, в моделируемой (а не в реальной) среде.

б) Целью тестирования является поиск расхождений между функциональной спецификацией и совокупностью программ ПС.

в) Целью тестирования является поиск несоответствия между описанием архитектуры и совокупностью программ ПС

Итоговая работа

1 часть

Таблица 1

Первая буква фамилии студента	№ варианта	№ практических заданий
А, Б	11.	1, 6
В, Г	12.	2, 7
Д, Е, Ё	13.	3, 8
Ж, З, И	14.	4, 9
К, Л, М	15.	5, 10
Н, О, П	16.	1, 11
Р, С, Т	17.	2, 12
У, Ф, Х	18.	3, 8
Ц, Ч, Ш, Щ	19.	4, 11
Э, Ю, Я	20.	5, 12

Практическое задание № 1

Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать два критерия выбора данных из массива и вывести эти данные на консоль.

Спецификация. *Сотрудник: id, ФИО, Должность, Стаж, Базовый оклад, Образование, Подразделение.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, реализация – в классе **EkzMain**.

Практическое задание № 2

Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать два критерия выбора данных из массива и вывести эти данные на консоль.

Спецификация. *Фильм: id, Название, Продолжительность, Жанр, Возрастное ограничение, Рейтинг, Режиссер.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, реализация – в классе **EkzMain**.

Практическое задание № 3

Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать два критерия выбора данных из массива и вывести эти данные на консоль.

Спецификация. *Блюдо (в ресторане): id, Название, Тип, Время приготовления, Калории, Размер порции в граммах, Стоимость.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, реализация – в классе **EkzMain**.

Практическое задание № 4

Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать два критерия выбора данных из массива и вывести эти данные на консоль.

Спецификация. Складское помещение: *id, Адрес, Завскладом, Общая площадь, Этажность, Аренда, Складское оборудование.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, реализация – в классе **EkzMain**.

Практическое задание № 5

Структура данных. Создать класс, спецификация которого приведена ниже. Определить конструктор и методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Задать два критерия выбора данных из массива и вывести эти данные на консоль.

Спецификация. Игрушка: *id, Название, Ограничение по возрасту, Производитель, Материал изготовления, Стоимость, Комплектность.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, реализация – в классе **EkzMain**.

Практическое задание № 6

Структура данных. Создать интерфейс и реализующий его класс (спецификация приведена ниже). Определить и реализовать методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Вывести данные из массива на консоль.

Спецификация. Призывник: *id, ФИО, Дата рождения, Результаты мед. освидетельствования.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, интерфейс – **EkzImpl**, реализация – в классе **EkzMain**.

Практическое задание № 7

Структура данных. Создать интерфейс и реализующий его класс (спецификация приведена ниже). Определить и реализовать методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Вывести данные из массива на консоль.

Спецификация. ПО: *id, Тип, Название, Лицензия, Стоимость.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, интерфейс – **EkzImpl**, реализация – в классе **EkzMain**.

Практическое задание № 8

Структура данных. Создать интерфейс и реализующий его класс (спецификация приведена ниже). Определить и реализовать методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Вывести данные из массива на консоль.

Спецификация. Город: *id, Название, Численность, Федеральный округ.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, интерфейс – **EkzImpl**, реализация – в классе **EkzMain**.

Практическое задание № 9

Структура данных. Создать интерфейс и реализующий его класс (спецификация приведена ниже). Определить и реализовать методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Вывести данные из массива на консоль.

Спецификация. Пользователь: *id, ФИО, Логин, Пароль.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, интерфейс – **EkzImpl**, реализация – в классе **EkzMain**.

Практическое задание № 10

Структура данных. Создать интерфейс и реализующий его класс (спецификация приведена ниже). Определить и реализовать методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Вывести данные из массива на консоль.

Спецификация. Соискатель: *id, ФИО, Должность, Стаж, Предыдущее место работы.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, интерфейс – **EkzImpl**, реализация – в классе **EkzMain**.

Практическое задание № 11

Структура данных. Создать интерфейс и реализующий его класс (спецификация приведена ниже). Определить и реализовать методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Вывести данные из массива на консоль.

Спецификация. Группа: *id, Наименование, Количество участников, Открытость, Тип.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, интерфейс – **EkzImpl**, реализация – в классе **EkzMain**.

Практическое задание № 12

Структура данных. Создать интерфейс и реализующий его класс (спецификация приведена ниже). Определить и реализовать методы **setТип()**, **getТип()**, **printInfo()**.

Реализация. Реализовать класс в основной программе. В основной программе создать списочный массив объектов. Вывести данные из массива на консоль.

Спецификация. Правонарушение: *id, Наименование, Вид, Ответственность, Нормативный правовой акт.*

Продемонстрировать подход, в котором реализуется **инкапсуляция** и обеспечивается **модульность**. Класс со структурой данных должен находиться в **EkzClass**, интерфейс – **EkzImpl**, реализация – в классе **EkzMain**.

2 часть

Таблица 2

Первая буква фамилии студента	№ варианта	Задание
А, Б	11.	По введенной дате рождения в поле формы приложения определить возраст человека
В, Г	12.	По введенным оценкам в поля формы приложения определить средний балл студента за сессию
Д, Е, Ё	13.	На форме приложения реализовать тест по трем вопросам
Ж, З, И	14.	По введенной стоимости товара и размеру скидки (в процентах) рассчитать стоимость покупки
К, Л, М	15.	По введенной стоимости товара и суммы транспортных расходов на доставку рассчитать итоговую стоимость
Н, О, П	16.	По введенной дате рождения в поле формы приложения определить подлежит человек призыву на срочную службу или нет
Р, С, Т	17.	По введенной дате в поле формы приложения определить месяц и вывести его наименование
У, Ф, Х	18.	По введенным оценкам студента в поля формы приложения определить количество оценок за сессию выше «3»
Ц, Ч, Ш, Щ	19.	По введенной в поле формы приложения дате поступления студента в ВУЗ определить продолжительность обучения на текущую дату
Э, Ю, Я	20.	На форме приложения реализовать анкету по трем вопросам

**Примерный перечень вопросов к зачету по дисциплине
"Объектно-ориентированное программирование"**

1. Основные принципы объектно-ориентированного программирования. Технологии и методы объектно-ориентированного программирования.
2. Объектно-ориентированный подход (ООП) к программированию.
3. Основные понятия ООП: объекты, классы и методы.
4. Редакции и реализации Java.
5. Принципы выбора оптимальных языков и инструментов разработки программ.
6. Связь Java с другими технологиями.
7. Программная среда: установка Java на ПК, программный инструментарий.
8. Лексика языка Java: кодировка, анализ программ, виды лексем.
9. Ключевые слова.
10. Литералы.
11. Операторы в языке Java.
12. Преобразование файлов различных кодировок.
13. Анализ программ.
14. Способы и средства преобразования файлов.
15. Инструменты анализа программ.
16. Java – типизированный язык.
17. Простые типы данных.
18. Преобразование типов данных в Java.
19. Понятие и виды ссылочных типов.
20. Классы, классы-оболочки.
21. Ссылочный строковый тип.
22. Определение классов в программе.
23. Массивы в Java.
24. Методы класса Arrays.
25. Многомерные массивы.
26. Создание и обработка массивов.
27. Конвенции кодирования.
28. Code Conventions for the Java (CCJ).
29. CCJ: структура, разделы. Отступы, комментарии.
30. CCJ: декларации, предложения языка, операторы.
31. Обработка исключений в Java.
32. Практика программирования, приемы программирования.
33. Классы: виды, элементы, объявление, инициализация.
34. Объявление классов, полей, методов. Перегруженные методы.
35. Конструкции классов и области видимости.
36. Объявление методов.
37. Перегруженные методы.
38. Создание классов, содержащих статические и нестатические методы.
39. Конструкторы. Объявление конструкторов.
40. Логика работы конструкторов. Ключевые слова super и this.
41. Инициализация, инициализаторы.
42. Определение цели и задач, разработки плана и целевых этапов создания программы.
43. Основные языки объектно-ориентированного программирования.

3Примерный перечень вопросов к экзамену по дисциплине "Объектно-ориентированное программирование"

1. Приведение типов классов.
2. Абстрактные классы и абстрактные методы.
3. Интерфейсы.
4. Интерфейсы на основе спецификаций классов.
5. Защищенный доступ к классам.
6. Пакеты.
7. Модификаторы доступа.
8. Создание интерфейсов.
9. Java Application Programming Interface. Ядро Java API.
10. Состав пакетов ядра Java API.
11. Типы Java-приложений.
12. Сравнение автономных приложений и апплетов.
13. Технология Java-JDBC.
14. Графические интерфейсы пользователя (GUI – для standalone applications или applets).
15. Анализ библиотек GUI.
16. Базовая архитектура AWT.
17. Пакет AWT: иерархия классов.
18. Апплеты: жизненный цикл, создание.
19. Класс Graphics: основные методы рисования.
20. Класс Graphics: работа с цветом, шрифтом.
21. Дополнительные методы. Класс FontMetrics.
22. Подклассы класса Component.
23. Контейнеры ScrollPane, Window, Frame, Dialog. Основные события.
24. Технология создания меню.
25. Библиотека Swing. JComponent.
26. Классы пакета javax.swing.
27. Схема MVC в компонентах Swing.
28. Создание окна JFrame. Панель содержимого, Container, JPanel.
29. Менеджеры размещения.
30. Особенности выравнивания элементов. Ручное и автоматическое размещение элементов.
31. События. Обработка событий.
32. Модель делегирования событий.
33. Семантические события.
34. Создание GUI-приложения.
35. Дженерик-классы. Дженерик-методы.
36. Типы коллекций.
37. Интерфейс Collection.
38. Иерархия коллекций.
39. Особенности использования коллекций.
40. Принципы объектно-ориентированного дизайна (ООД) классов.
41. Организация доступа к базе данных из приложения.
42. Практические приемы разработки приложений на объектно-ориентированном языке.