

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ



Декан факультета ИВТ

Д.Ю. Чалый

(подпись)

« 18 » _____ мая _____ 2021 ____ г.

**Рабочая программа дисциплины
«Информационные системы и технологии»**

Направление подготовки
09.03.03 Прикладная информатика

Направленность (профиль)
«Информационные технологии в цифровой экономике»

Форма обучения
очная

Программа одобрена
на заседании кафедры
от «23» апреля 2021 года, протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 7 от « 17 » мая 2021года

Ярославль
2021

1. Цели освоения дисциплины

Целями дисциплины «Информационные системы и технологии» являются изучение основных этапов, методов, средств и стандартов разработки программного обеспечения, детальное изучение одного из объектно-ориентированных языков программирования (Java), библиотек для создания графических пользовательских интерфейсов, обучение студентов разработке кросс-платформенных приложений.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Информационные системы и технологии» относится к обязательной части ОП бакалавриата.

Освоению данной программы предшествуют учебные курсы по программированию и современным информационным технологиям.

Дисциплина «Информационные системы и технологии» обеспечивает закрепление и углубление теоретических знаний и практических навыков по основным дисциплинам ИТ-цикла. Дисциплина способствует профессиональному росту студентов, повышению их общеметодологического уровня, а также дальнейшему развитию навыков научно-исследовательской деятельности.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ООП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		
ООПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;	ООПК-2.1 Обладает навыками выбора современных программных средств для решения прикладных задач в профессиональной деятельности.	Знать: <ul style="list-style-type: none">• основы теории алгоритмов и ее применения;• принципы объектно-ориентированного программирования. Уметь: <ul style="list-style-type: none">• разрабатывать программы на языке Java;• выполнять отладку и тестирование программы. Владеть навыками: <ul style="list-style-type: none">• работы с интегрированной средой разработки NetBeans;• работы с научно-технической литературой и технической документацией Java и JavaFX.

<p>ОПК-8. Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла;</p>	<p>ОПК-8.1 демонстрирует способность принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла;</p>	<p>Знать:</p> <ul style="list-style-type: none"> • основы теории алгоритмов и ее применения; • синтаксис языка программирования Java; • принципы объектно-ориентированного программирования. <p>Уметь:</p> <ul style="list-style-type: none"> • разрабатывать программы на языке Java; • создавать графические пользовательские приложения с использованием JavaFX; • разрабатывать приложения в рамках архитектуры «Модель-Вид-Контроллер». <p>Владеть навыками:</p> <ul style="list-style-type: none"> • работы с интегрированной средой разработки NetBeans; • работы с научно-технической литературой и технической документацией Java и JavaFX.
---	---	--

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зачетных единиц, 108 акад. часов.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам) <i>Формы ЭО и ДОТ (при наличии)</i>
			Контактная работа					самостоятельная работа	
			лекции	практические	лабораторные	консультации	аттестационные испытания		
1	Объектно-ориентированное программирование. Основные понятия.	3	0,5	2				2	Задания для самостоятельной работы
2	Синтаксис языка Java	3	0,5	2				2	Контрольная работа №1
	<i>в том числе ЭО и ДОТ</i>							2	
3	Классы. Элементы класса.	3	1	2				2	Лабораторная работа
4	Управление доступом. Инкапсуляция.	3	1	2				2	Задания для самостоятельной работы
5	Наследование и полиморфизм	3	1	2				2	
	<i>в том числе ЭО и ДОТ</i>							2	
6	Интерфейсы	3	1	2				2	
7	Обработка исключительных ситуаций.	3	1	2				2	Контрольная работа №2
8	Основные классы для ввода/вывода.	3	1	2				2	Задания для самостоятельной работы
	<i>в том числе ЭО и ДОТ</i>							3	
9	Обработка текстовой информации с использованием класса String.	3	1	2				2	Лабораторная работа
10	Контейнеры, коллекции	3	1	2				2	
11	Графические пользовательские приложения	3	1	2				2	
12	Введение в JavaFX	3	1	2				2	Контрольная работа №3
	<i>в том числе ЭО и ДОТ</i>							3	
13	Архитектура создания пользовательских приложений: Модель Вид Контроллер.	3	1	2				2	Задания для самостоятельной работы
14	Основные элементы сцены	3	1	2				2	Лабораторная работа
15	Взаимодействие между элементами приложения. События	3	1	2				2	

16	Организация ввода данных. Диалоговые окна.	3	1	1			3	Задания для самостоятельной работы
17	Главное окно приложения.	3	1	1			2	
	<i>в том числе ЭО и ДОТ</i>						3	
18	Разработка декларативного интерфейса с помощью FXML	3	1	2			3,7	
		3						Зачет
	ИТОГО		17	34			0,3	51,7
	<i>в том числе с ЭО и ДОТ</i>						13	

Содержание разделов дисциплины:

1. Объектно-ориентированное программирование. Основные понятия. Объект, класс, метод. Инкапсуляция. Наследование. Полиморфизм.
2. Синтаксис языка Java. Базовые конструкции языка Java. Переменные. Типы данных, преобразование типов.
3. Классы. Элементы класса. Поля и методы. Конструкторы. Пример разработки пользовательского класса Point.
4. Управление доступом. Инкапсуляция. Доступ к членам класса.
5. Наследование и полиморфизм. Базовые и производные классы. Доступ к элементам класса. Переопределение методов. Абстрактные классы. Примеры
6. Интерфейсы. Определение интерфейса. Реализация интерфейса. Стандартные интерфейсы.
7. Обработка исключительных ситуаций. Синтаксис, ключевые слова. Типы исключений. Общая схема работы с исключительными ситуациями: генерация исключений, перехват исключений, блоки-ловушки.
8. Основные классы для ввода/вывода. Иерархия классов. Работа с файлами.
9. Обработка текстовой информации с использованием класса String. Основные методы класса, операции со строками.
10. Контейнеры, коллекции. Основные классы и интерфейсы. Итераторы. Последовательности и множества. Свойства и преимущества контейнеров.
11. Графические пользовательские приложения. Инструменты разработки: фреймворк, среда разработки, система контроля версий.
12. Введение в JavaFX. Понятие графической библиотеки. Документация.
13. Архитектура создания пользовательских приложений: Модель Вид Контроллер. Основные элементы архитектурного шаблона. Связи между элементами.
14. Основные элементы сцены. Элементы отображения, элементы управления, невидимые элементы. Свойства элементов. Компановка элементов.
15. Взаимодействие между элементами приложения. События. Обработчики событий. Лямбда-выражения.
16. Организация ввода данных. Диалоговые окна. Стандартные диалоги. Создание пользовательских диалоговых окон..
17. Главное окно приложения. Меню, виды меню. Расположение элементов. Обработка событий.
18. Разработка декларативного интерфейса с помощью FXML. Реализация элементов архитектуры «Модель Вид Контроллер».

5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Лекция-беседа или «диалог с аудиторией», является наиболее распространенной и сравнительно простой формой активного вовлечения студентов в учебный процесс. Эта лекция предполагает непосредственный контакт преподавателя с аудиторией. Преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание студентов к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей студентов.

Мастер-класс – это особая форма учебного занятия, когда преподаватель-мастер передает свой опыт путем прямого и комментированного показа последовательности действий, методов, приемов и форм педагогической деятельности. Целью проведения мастер-класса является профессиональное, интеллектуальное и эстетическое воспитание студентов, и прежде всего, развитие в ходе мастер-класса способности студента самостоятельно и нестандартно мыслить.

В процессе обучения используются следующие технологии электронного обучения и дистанционные образовательные технологии:

Электронный учебный курс «Информационные системы и технологии» в LMS Электронный университет Moodle ЯрГУ, в котором:

- представлены задания для самостоятельной работы обучающихся по темам дисциплины;
- осуществляется проведение отдельных мероприятий текущего контроля успеваемости студентов;
- представлены тексты лекций по отдельным темам дисциплины;
- представлены правила прохождения промежуточной аттестации по дисциплине;
- представлен список учебной литературы, рекомендуемой для освоения дисциплины;
- представлена информация о форме и времени проведения консультаций по дисциплине в режиме онлайн;
- посредством форума осуществляется синхронное и (или) асинхронное взаимодействие между обучающимися и преподавателем в рамках изучения дисциплины.

6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине

В процессе осуществления образовательного процесса по дисциплине используются:

для формирования материалов для текущего контроля успеваемости и проведения промежуточной аттестации, для формирования методических материалов по дисциплине:

- программы Microsoft Office;
- издательская система LaTeX;
- для выполнения лабораторных работ среда разработки NetBeans 8 (свободная)

7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

В процессе осуществления образовательного процесса по дисциплине используются:

Автоматизированная библиотечно-информационная система «БУКИ-NEXT»
http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php

8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины

а) основная литература

1. Васильев, А. Н., Java. Объектно-ориентированное программирование для магистров и бакалавров : базовый курс по объектно-ориентированному программированию / А. Н. Васильев, СПб., Питер, 2014, 396с
2. Лагутина, Н. С., Разработка программных приложений : практикум для студентов, обучающихся по направлению Фундаментальная информатика и информационные технологии / Н. С. Лагутина, Ю. А. Ларина, А. М. Васильев; Яросл. гос. ун-т., Ярославль, ЯрГУ, 2014, 71 с.
3. Лагутина, Н. С., Разработка программных приложений [Электронный ресурс] : практикум для студентов, обучающихся по направлению Фундаментальная информатика и информационные технологии / Н. С. Лагутина, Ю. А. Ларина, А. М. Васильев; Яросл. гос. ун-т., Ярославль, ЯрГУ, 2014, 71 с.

б) дополнительная литература

1. Парамонов, И. В., Язык программирования Java и Java - технологии : учеб. пособие / И. В. Парамонов ; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2006, 91с
2. Парамонов, И. В., Язык программирования Java и Java - технологии [Электронный ресурс] : учеб. пособие / И. В. Парамонов ; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2006, 91с
3. Гонсалвес, Э., Изучаем Java EE 7 / Э. Гонсалвес; [пер. с англ. Е. Зазноба и др.], СПб., Питер, 2016, 640с
4. Эккель, Б., Философия Java / Б. Эккель; пер. с англ. Е. Матвеева. - 4-е полное изд., М., Питер, 2017, 1165с

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа;
- учебные аудитории для проведения практических занятий (семинаров);
- учебные аудитории для проведения групповых и индивидуальных консультаций;
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде ЯрГУ.

Автор:

Доцент кафедры
вычислительных и программных
систем, к.ф.-м.н.

должность, ученая степень

подпись

Н.С. Лагутина

И.О. Фамилия

**Приложение № 1 к рабочей программе дисциплины
«Информационные системы и технологии»**

**Фонд оценочных средств
для проведения текущего контроля успеваемости
и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания и иные материалы,
используемые в процессе текущего контроля успеваемости**

**1.1. Контрольные задания и иные материалы, используемые в процессе текущей
аттестации**

Задания для самостоятельной работы

Примеры заданий по теме «Синтаксис языка Java»:

1. Вводится натуральное число N. Представьте его в виде суммы степеней двойки.
Результат напечатать.

Возможные тесты

Входные данные	Результат
5	1+4
255	1+2+4+8+16+32+64+128
1	1
0	Введено не натуральное число

2. В массиве целых чисел найти наиболее часто встречающееся число. Если таких чисел несколько, то определить наименьшее из них.

Возможные тесты

Входные данные	Результат
1 2 3 4	1
2 5 7 8 13 2 5 7 8 2 5 6	2
2 1 2	2
6	6

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-8: Знает и применяет подходящие для решения алгоритмы, выбирает наиболее эффективный алгоритм. Знает и применяет синтаксические конструкции языка Java. Программа выполняется для всех тестов ОПК-2: Создает полноценное консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости
Хорошо	ОПК-8: Знает и применяет подходящие для решения стандартные

	алгоритмы. Знает и применяет синтаксические конструкции языка Java. Программа выполняется для всех тестов, если какой-либо из тестов не выполняется программа дорабатывается. ОПК-2: Создает консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости с небольшими неточностями
Удовлетворительно	ОПК-8: Знает и применяет простые алгоритмы. Знает и применяет синтаксические конструкции языка Java. Программа выполняется для всех тестов, возможно кроме одного-двух крайних случаев. ОПК-2: Создает консольное приложение в среде разработки. С трудом поясняет код, не может изменить код при усложнении или существенном дополнении задачи
Неудовлетворительно	ОПК-8: Не может подобрать подходящие для решения алгоритмы. Ошибается во многих синтаксических конструкциях языка Java. Программа не выполняется для большинства тестов. ОПК-2: Не может создать консольное приложение в среде разработки или создает неработающий проект. Не может пояснить код и изменить его.

Примеры заданий по теме «Наследование и полиморфизм»:

- Опишите модель данных для вычисления заработной платы работников организации: работающих постоянно (их заработная плата определяется размером должностного оклада и дополнительной премией в процентах от оклада) и работающих по договору (их заработная плата определяется суммой денег, выплачиваемых за один день работы и количеством отработанных дней). Необходимо вывести на экран зарплату инженера с окладом 20 000 рублей и премией 40 %, а также дизайнера, проработавшего 12 дней с оплатой 950 рублей в день. Разработать модель данных, основанную на иерархии классов. Базовый класс описывает общие поля и методы работников, в классах наследниках добавляются собственные свойства и описывается поведение.
- Опишите модель данных для вычисления площадей полной поверхности и объемов геометрических фигур в пространстве: конусов ($S = \pi R^2 + \pi R L$, $V = 1/3 \pi R^2 H$) и прямоугольных параллелепипедов. Необходимо вывести на экран площадь полной поверхности и объем конуса с радиусом основания 3 и высотой 4, а также прямоугольного параллелепипеда с длиной 5, шириной 4 и высотой 3. Разработать модель данных, основанную на иерархии классов. Базовый класс описывает общие поля и методы фигур, в классах наследниках добавляются собственные свойства и описывается поведение.

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-8: Знает принципы объектно-ориентированного программирования. Разрабатывает соответствующий задаче класс или классы для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы. Каждый метод решает только одну задачу. Среди аргументов методов нет лишних. Умеет создавать объекты классов и работать с ними. ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса. Создает полноценное консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости.

Хорошо	ОПК-8: Знает принципы объектно-ориентированного программирования. Разрабатывает соответствующий задаче класс или классы для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы. Каждый метод решает только одну задачу, возможно кроме одного-двух спроектированных недостаточно верно. Умеет создавать объекты классов и работать с ними. ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для одного-двух методов неверно назначен уровень доступа. Создает полноценное консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости с небольшими неточностями
Удовлетворительно	ОПК-8: Знает принципы объектно-ориентированного программирования. Разрабатывает соответствующий задаче класс или классы для хранения и обработки данных. В классе содержатся необходимые поля, но могут присутствовать лишние. Присутствуют не все необходимые методы или добавлены явно лишние. Некоторые методы спроектированных недостаточно верно. Умеет создавать объекты классов и работать с ними. ОПК-2: Не вполне понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для ряда полей и методов неверно назначен уровень доступа. Создает полноценное консольное приложение в среде разработки. С трудом поясняет и изменяет код.
Неудовлетворительно	ОПК-8: Не знает или плохо знает принципы объектно-ориентированного программирования. Не может разработать соответствующий задаче класс для хранения и обработки данных. Не может создать метод класса.. Не умеет создавать и работать с объектами классов. ОПК-2: Не понимает принципы инкапсуляции и правила определения доступа к элементам класса. Не может создать консольное приложение в среде разработки или создает неработающий проект. Не может пояснить код и изменить его.

Примеры заданий по теме «Обработка текстовой информации с использованием класса String»:

1. Вводится предложение, слова разделены пробелами. Расположить слова в нём в порядке возрастания числа букв в словах.

Возможные тесты

Входные данные	Результат
He likes classical music	He likes music classical
They are on holiday	on are They holiday
Are on	on Are
on	on

2. Вводится текст телеграммы и стоимость одного слова. Слова разделены пробелами, знаки препинания считаются отдельными словами. Определить стоимость телеграммы, если цена одного слова 2 руб..

Возможные тесты

Входные данные	Результат
В случае неявки, акт осмотра будет составлен без Вашего участия.	24
Текст телеграммы	4
телеграмма	2
«пустая строка»	0 или сообщение об отсутствии текста

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-8: Знает класс String и применяет методы этого класса, подходящие для решения задачи. Знает и применяет регулярные выражения. Программа выполняется для всех тестов ОПК-2: Создает полноценное консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости
Хорошо	ОПК-8: Знает класс String и применяет методы этого класса, подходящие для решения задачи в некоторых случаях с подсказкой преподавателя. Знает и применяет регулярные выражения. Программа выполняется для всех тестов, если какой-либо из тестов не выполняется программа дорабатывается. ОПК-2: Создает консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости с небольшими неточностями
Удовлетворительно	ОПК-8: Знает класс String, но не может без подсказки выбрать методы и полностью самостоятельно составить алгоритм для решения задачи либо решает только самые простые задачи. Знает регулярные выражения., но составляет их с ошибками и не всегда понимает в каких случаях они применяются Программа выполняется для всех тестов, возможно кроме одного-двух крайних случаев. ОПК-2: Создает консольное приложение в среде разработки. С трудом поясняет код, не может изменить код при усложнении или существенном дополнении задачи
Неудовлетворительно	ОПК-8: Не умеет создавать и использовать объекты класса String. Не может подобрать подходящие для решения методы этого класса. Не может составить регулярное выражение. Программа не выполняется для большинства тестов. ОПК-2: Не может создать консольное приложение в среде разработки или создает неработающий проект. Не может пояснить код и изменить его.

Примеры заданий по теме «Основные элементы сцены»

1. Разработать простое графическое приложение, состоящее только из одного окна, содержащего виджеты для ввода и вывода данных. Определить стоимость весового товара. Для каждого товара известна цена одного килограмма в рублях. Окно приложения содержит поле для выбора товара (не менее 20 вариантов), поле для ввода веса в граммах и кнопку «ОК». После нажатия на кнопку появляется сообщение о стоимости товара.
2. Разработать простое графическое приложение, состоящее только из одного окна, содержащего виджеты для ввода и вывода данных. Определить флаг заданной страны. Окно приложения содержит поле для выбора или ввода названия страны (не менее 30

вариантов), виджет для отображения картинки с флагом и кнопку «ОК». После нажатия на кнопку в виджете появляется соответствующий рисунок или выдаётся сообщение о том, что информации о флаге страны нет.

Критерии оценивания

Оценка	Критерии
Отлично	<p>ОПК-8: Знает принципы разработки графических пользовательских приложений с использованием JavaFX. Умеет выбрать и разместить подходящие визуальные элементы сцены окна приложения. Знает параметры настройки этих элементов. Умеет работать с текстовыми и графическими файлами как с ресурсами проекта. Знает и использует способы изменения внешнего вида элементов графического интерфейса.</p> <p>ОПК-2: Умеет создавать однооконное графическое приложение в среде разработки. Поясняет код и изменяет его при необходимости. Использует техническую документацию при разработке программы, материалы лекций и другие источники информации.</p>
Хорошо	<p>ОПК-8: Знает принципы разработки графических пользовательских приложений с использованием JavaFX. Умеет выбрать и разместить подходящие визуальные элементы сцены окна приложения с небольшими подсказками преподавателя. Знает отдельные параметры настройки этих элементов. Умеет работать с текстовыми и графическими файлами как с ресурсами проекта. Знает и использует некоторые способы изменения внешнего вида элементов графического интерфейса.</p> <p>ОПК-2: Умеет создавать однооконное графическое приложение в среде разработки. Поясняет код и изменяет его при необходимости с небольшими неточностями. Иногда использует техническую документацию при разработке программы, в основном пользуется лекционными материалами.</p>
Удовлетворительно	<p>ОПК-8: Знает принципы разработки графических пользовательских приложений с использованием JavaFX. Умеет выбрать и разместить подходящие визуальные элементы сцены окна приложения с подсказками преподавателя. Не умеет изменять параметры настройки элементов сцены или изменяет только некоторые параметры. Умеет работать с текстовыми файлами. Не знает или использует только некоторые способы изменения внешнего вида элементов графического интерфейса.</p> <p>ОПК-2: Умеет создавать однооконное графическое приложение в среде разработки. С трудом поясняет и изменяет код. Плохо ориентируется в технической документации и в лекционных материалах.</p>
Неудовлетворительно	<p>ОПК-8: Не знает принципы разработки графических пользовательских приложений. Не умеет выбрать и разместить подходящие визуальные элементы сцены окна приложения. Не умеет изменять параметры настройки элементов сцены. Не знает способы изменения внешнего вида элементов графического интерфейса.</p> <p>ОПК-2: Не умеет создавать однооконное графическое</p>

	приложение в среде разработки или создает неработающий проект. Не может пояснить код и изменить его. Не ориентируется в технической документации и в лекционных материалах.
--	---

Примеры заданий по теме «Главное окно приложения»:

1. Используя архитектуру «Модель-Вид-Контроллер» разработать приложение, отображающее данные о треугольнике:

Модель – класс, хранящий и вычисляющий информацию о треугольнике.

Вид (1) – виджет, в котором нарисован треугольник.

Вид (2) – виджет, отображающий информацию в текстовом виде (параметры треугольника, его площадь и периметр).

Контроллер – главное окно приложения, в котором вводится информация о треугольнике (длины сторон), происходит проверка вводимых данных и вызов виджетов, отображающих объект.

Разработать способ хранения данных (в векторе, списке, множестве и т.п.), организовать возможность добавления данных, изменения имеющихся данных, отображения всех введенных данных. Реализовать возможность чтения данных из текстового файла. Все вводимые данные должны быть проверены на правильность.

2. Используя архитектуру «Модель-Вид-Контроллер» разработать приложение позволяющее работать с данными о студенте. Данные содержат:

- фамилию, имя, отчество;
- пол
- возраст
- номер группы
- предмет
- сведения об оценке

Модель – класс, хранящие информацию о студенте.

Вид – виджет, в котором отображаются сведения о студенте.

Контроллер – главное окно приложения, в котором находится меню, содержащее пункты, необходимые для ввода и обработки данных.

Разработать способ хранения данных (в векторе, списке, множестве и т.п.), организовать возможность добавления данных, изменения имеющихся данных, отображения всех введенных данных. Реализовать возможность чтения данных из текстового файла. Все вводимые данные должны быть проверены на правильность.

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-8: Знает принципы разработки многооконных графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Четко разделяет классы программы в соответствии с этой архитектурой. Умеет построить связи между частями приложения. Может использовать дополнительные шаблоны программирования. Реализует алгоритмы выполнения функций приложения в соответствии с поставленной задачей. Умеет протестировать все сценарии использования программы. ОПК-2: Умеет создавать многооконное графическое приложение в среде разработки. Знает и использует необходимые графические элементы, в том числе

	<p>соответствующие стандартам разработки пользовательских приложений: меню, диалоговые окна, различные элементы отображения и ввода информации. Поясняет код и изменяет его при необходимости. Использует техническую документацию при разработке программы, материалы лекций и другие источники информации.</p>
Хорошо	<p>ОПК-8: Знает принципы разработки многооконных графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Четко разделяет классы программы в соответствии с этой архитектурой. Умеет построить связи между частями приложения, в некоторых случаях с помощью преподавателя. Реализует алгоритмы выполнения функций приложения в соответствии с поставленной задачей. Умеет протестировать почти все сценарии использования программы.</p> <p>ОПК-2: Умеет создавать многооконное графическое приложение в среде разработки. Знает и использует необходимые графические элементы, в том числе соответствующие стандартам разработки пользовательских приложений: меню, диалоговые окна, различные элементы отображения и ввода информации. Поясняет код и изменяет его при необходимости с небольшими неточностями. Иногда использует техническую документацию при разработке программы, в основном пользуется лекционными материалами.</p>
Удовлетворительно	<p>ОПК-8: Знает принципы разработки многооконных графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Но нечетко разделяет классы программы в соответствии с этой архитектурой. Не всегда может правильно построить связи между частями приложения. Реализует большую часть алгоритмов выполнения функций приложения в соответствии с поставленной задачей. Тестирует некоторые сценарии использования программы.</p> <p>ОПК-2: Умеет создавать многооконное графическое приложение в среде разработки. Знает и использует основные графические элементы, рассмотренные на лекциях. С трудом поясняет и изменяет код. Плохо ориентируется в технической документации и в лекционных материалах.</p>
Неудовлетворительно	<p>ОПК-8: Не знает принципы разработки многооконных графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Не может разделить классы программы в соответствии с этой архитектурой. Не может правильно построить связи между частями приложения. Не может реализовать большую часть алгоритмов выполнения функций приложения. Не знает принципы тестирования графических приложений.</p> <p>ОПК-2: Не умеет создавать графическое приложение в</p>

	среде разработки или создает неработающий проект. Не знает стандарты разработки графических пользовательских приложений. Не может пояснить код и изменить его. Не ориентируется в технической документации и в лекционных материалах.
--	---

Контрольная работа №1

Вариант 1

Сущность "трехмерный вектор" характеризуется координатами (три действительных числа). Создайте класс для хранения и обработки данных о векторе, где есть методы позволяющие определить модуль вектора, вывести информацию о векторе в текстовом виде, вычислить сумму двух векторов (покоординатно);

Вариант 2

Сущность "человек" характеризуется именем и годом рождения (строка и целое число). Создайте класс для хранения и обработки данных о человеке, где есть метод, определяющий возраст человека в заданном году, метод, сравнивающий двух людей на равенство (совпадение имени и года рождения), метод, выводящий информацию о человеке в текстовом виде;

Вариант 3

Сущность "круг" характеризуется координатами центра и радиусом (три действительных числа). Создайте класс для хранения и обработки данных о круге, где есть методы позволяющие определить площадь круга, вывести информацию о круге в текстовом виде, сравнить два круга на равенство (круги равны, если совпадают их центры и радиусы);

Вариант 4

Сущность "абитуриент" характеризуется именем и суммой баллов (строка и целое число). Создайте класс для хранения и обработки данных об абитуриенте, где есть метод, определяющий проходит ли абитуриент конкурс при заданном проходном балле, метод, сравнивающий двух абитуриентов по количеству баллов, метод, выводящий информацию об абитуриенте в текстовом виде;

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-8: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы, все алгоритмы реализованы верно. Каждый метод решает только одну задачу. Среди аргументов методов нет лишних. Умеет создавать объекты классов и работать с ними. ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса.
Хорошо	ОПК-8: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы, алгоритмы реализованы верно возможно с небольшими неточностями. Каждый метод решает только одну задачу, возможно кроме одного-двух спроектированных недостаточно верно. Умеет создавать объекты классов и работать с ними. ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для

	одного-двух методов неверно назначен уровень доступа.
Удовлетворительно	ОПК-8: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся необходимые поля, но могут присутствовать лишние. Присутствуют не все необходимые методы или добавлены явно лишние. Алгоритмы реализации методов содержат ошибки, но не более двух грубых. Некоторые методы спроектированных недостаточно верно. Умеет создавать объекты классов и работать с ними. ОПК-2: Не вполне понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для ряда полей и методов неверно назначен уровень доступа.
Неудовлетворительно	ОПК-8: Не знает или плохо знает принципы объектно-ориентированного программирования. Не может разработать соответствующий задаче класс для хранения и обработки данных. Не может создать метод класса, ошибается в алгоритмах. Не умеет создавать и работать с объектами классов. ОПК-2: Не понимает принципы инкапсуляции и правила определения доступа к элементам класса.

Контрольная работа №2

Вариант 1

Сущность "трехмерный вектор" характеризуется координатами (три действительных числа).

- а) создайте класс для хранения и обработки данных о векторе, где есть методы позволяющие определить модуль вектора, вывести информацию о векторе в текстовом виде, вычислить сумму двух векторов (покоординатно);
- б) в файле с именем data.txt содержится информация о двух векторах (шесть чисел); создайте два соответствующих объекта, выведите на консоль информацию о каждом векторе, его модуль, а также сумму этих двух векторов.

Вариант 2

Сущность "человек" характеризуется именем и годом рождения (строка и целое число).

- а) создайте класс для хранения и обработки данных о человеке, где есть метод, определяющий возраст человека в заданном году, метод, сравнивающий двух людей на равенство (совпадение имени и года рождения), метод, выводящий информацию о человеке в текстовом виде;
- б) в файле с именем data.txt содержится информация о двух людях (строка число строка число); создайте два соответствующих объекта, выведите на консоль информацию о каждом человеке, его возраст в текущем году, а также информацию о сравнении этих двух людей.

Вариант 3

Сущность "круг" характеризуется координатами центра и радиусом (три действительных числа).

- а) создайте класс для хранения и обработки данных о круге, где есть методы позволяющие определить площадь круга, вывести информацию о круге в текстовом виде, сравнить два круга на равенство (круги равны, если совпадают их центры и радиусы);
- б) в файле с именем data.txt содержится информация о двух кругах (шесть чисел); создайте два соответствующих объекта, выведите на консоль информацию о каждом круге, его площадь, а также информацию о сравнении этих двух кругов.

Вариант 4

Сущность "абитуриент" характеризуется именем и суммой баллов (строка и целое число).

а) создайте класс для хранения и обработки данных об абитуриенте, где есть метод, определяющий проходит ли абитуриент конкурс при заданном проходном балле, метод, сравнивающий двух абитуриентов по количеству баллов, метод, выводящий информацию об абитуриенте в текстовом виде;

б) в файле с именем data.txt содержится информация о двух абитуриентах (строка число строка число); создайте два соответствующих объекта, выведите на консоль информацию о каждом абитуриенте в порядке убывания количества баллов, а также проходят ли они конкурс при проходном балле 200.

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-8: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы, все алгоритмы реализованы верно. Каждый метод решает только одну задачу. Среди аргументов методов нет лишних. Выполняет ввод и вывод данных, умеет работать с файлами. Реализует подходящие интерфейсы. ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса. Использует способы сравнения объектов, определенные стандартом языка Java (реализация стандартных интерфейсов и использование стандартных алгоритмов).
Хорошо	ОПК-8: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы, алгоритмы реализованы верно возможно с небольшими неточностями. Каждый метод решает только одну задачу, возможно кроме одного-двух спроектированных недостаточно верно. Выполняет ввод и вывод данных, умеет работать с файлами. Реализует подходящие интерфейсы с небольшими неточностями. ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для одного-двух методов неверно назначен уровень доступа. Использует способы сравнения объектов, определенные стандартом языка Java (реализация стандартных интерфейсов и использование стандартных алгоритмов), возможно не во всех случаях.
Удовлетворительно	ОПК-8: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся необходимые поля, но могут присутствовать лишние. Присутствуют не все необходимые методы или добавлены явно лишние. Алгоритмы реализации методов содержат ошибки, но не более двух грубых. Некоторые методы спроектированных недостаточно верно. Выполняет ввод и вывод данных с ошибками. Реализует подходящие интерфейсы, но с ошибками.

	ОПК-2: Не вполне понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для ряда полей и методов неверно назначен уровень доступа. Использует способы сравнения объектов, определенные стандартом языка Java (реализация стандартных интерфейсов и использование стандартных алгоритмов) с ошибками или реализует собственные методы вместо использования стандартных.
Неудовлетворительно	ОПК-8: Не знает или плохо знает принципы объектно-ориентированного программирования. Не может разработать соответствующий задаче класс для хранения и обработки данных. Не может создать метод класса, ошибается в алгоритмах. Не умеет выполнять ввод и вывод данных. ОПК-2: Не понимает принципы инкапсуляции и правила определения доступа к элементам класса. Не умеет реализовывать стандартные интерфейсы и использовать стандартные алгоритмы.

Контрольная работа №3

Вариант 1

В рамках архитектуры «Модель-Вид-Контроллер» разработайте классы для хранения, изменения и отображения данных о товаре. Товар обладает ценой, названием, сроком годности.

Вариант 2

В рамках архитектуры «Модель-Вид-Контроллер» разработайте классы для хранения, изменения и отображения времени. Информация о времени – часы и минуты.

Вариант 3

В рамках архитектуры «Модель-Вид-Контроллер» разработайте классы для калькулятора, умеющего складывать и вычитать два числа. Числа и знак действия вводит пользователь.

Вариант 4

В рамках архитектуры «Модель-Вид-Контроллер» разработайте классы для решения квадратного уравнения. Коэффициенты уравнения вводит пользователь.

Вариант 5

В рамках архитектуры «Модель-Вид-Контроллер» разработайте классы для решения задачи: пользователь вводит координаты трёх точек на оси ОХ: А, В, С. Определить, какая из точек В или С расположена ближе к А.

Вариант 6

В рамках архитектуры «Модель-Вид-Контроллер» разработайте классы для решения задачи: осуществить перевод величин из радианной меры в градусную и наоборот.

Критерии оценивания

Оценка	Критерии
Отлично	<p>ОПК-8: Знает принципы разработки графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Четко разделяет классы программы в соответствии с этой архитектурой. Умеет построить связи между частями приложения. Реализует алгоритмы выполнения функций приложения в соответствии с поставленной задачей.</p> <p>ОПК-2: Знает и использует необходимые графические элементы, в том числе соответствующие стандартам разработки пользовательских приложений: элементы отображения и ввода информации.</p>
Хорошо	<p>ОПК-8: Знает принципы разработки многооконных графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Четко разделяет классы программы в соответствии с этой архитектурой. Умеет построить связи между частями приложения с некоторыми неточностями или небольшими ошибками. Реализует алгоритмы выполнения функций приложения в соответствии с поставленной задачей.</p> <p>ОПК-2: Знает и использует необходимые графические элементы, в том числе соответствующие стандартам разработки пользовательских приложений: элементы отображения и ввода информации, но допускает синтаксические ошибки.</p>
Удовлетворительно	<p>ОПК-8: Знает принципы разработки графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Но нечетко разделяет классы программы в соответствии с этой архитектурой. Не всегда может правильно построить связи между частями приложения. Реализует большую часть алгоритмов выполнения функций приложения в соответствии с поставленной задачей, но не все или с существенными ошибками.</p> <p>ОПК-2: Знает и использует основные графические элементы, рассмотренные на лекциях. Допускает много синтаксических ошибок, некоторые элементы использует некорректно.</p>
Неудовлетворительно	<p>ОПК-8: Не знает принципы разработки многооконных графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Не может разделить классы программы в соответствии с этой архитектурой. Не может правильно построить связи между частями приложения. Не может реализовать большую часть алгоритмов выполнения функций приложения.</p> <p>ОПК-2: Не знает стандарты разработки графических пользовательских приложений. Не знает синтаксис элементов графического приложения и их методов.</p>

Задания для лабораторных работ

Лабораторная работа по теме «Управление доступом. Инкапсуляция.»

Разработать подходящие классы для решения задачи: вовремя ли читатель сдал книгу в библиотеку?

Поля класса являются закрытыми. В классе должны быть: конструктор по умолчанию, конструктор с параметрами, конструктор копирования, методы возвращающие и изменяющие значения полей, методы для решения задачи. Для проверки вовремя ли сдана книга надо написать отдельный метод (подумайте в каком классе он должен располагаться и какие параметры должны быть у этого метода).

В главном классе приложения создайте двух читателей: Петров, взявший книгу Г. Шилдта «Java 8. Полное руководство» до 1 апреля и Васечкин, взявший книгу Л. Толстого «Крейцера соната» до 25 марта и определите кто сдал книгу вовремя, если оба сдали книги 29 марта.

Подсказка.

Класс книга: возможные поля – название, автор, день сдачи книги, номер месяца сдачи книги.

Класс читатель: возможные поля – имя, книга, взятая в библиотеке.

Другой вариант.

Класс книга: возможные поля – название, автор.

Класс читатель: возможные поля – имя, книга, взятая в библиотеке, день сдачи книги, номер месяца сдачи книги.

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-8: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы, все алгоритмы реализованы верно. Каждый метод решает только одну задачу. Среди аргументов методов нет лишних. ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса.
Хорошо	ОПК-8: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы, алгоритмы реализованы верно возможно с небольшими неточностями. Каждый метод решает только одну задачу, возможно кроме одного-двух спроектированных недостаточно верно. ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для одного-двух методов неверно назначен уровень доступа.
Удовлетворительно	ОПК-8: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся необходимые поля, но могут присутствовать лишние. Присутствуют не все необходимые методы или добавлены явно лишние. Алгоритмы реализации методов содержат ошибки, но не более двух грубых. Некоторые методы спроектированных недостаточно верно. ОПК-2: Не вполне понимает принципы инкапсуляции и правила определения доступа к элементам класса,

	возможно для ряда полей и методов неверно назначен уровень доступа.
Неудовлетворительно	ОПК-8: Не знает или плохо знает принципы объектно-ориентированного программирования. Не может разработать соответствующий задаче класс для хранения и обработки данных. Не может создать метод класса, ошибается в алгоритмах. ОПК-2: Не понимает принципы инкапсуляции и правила определения доступа к элементам класса.

Задания по теме. «Контейнеры, коллекции»

1. Создать динамический массив целых чисел (`ArrayList<Integer>`) и решить следующие задачи:

- удалить все отрицательные числа;
- определить номер максимального элемента;
- отсортировать числа по возрастанию;
- определить, есть ли в массиве простые числа.

2. Создать список действительных чисел (`LinkedList<Double>`) и решить следующие задачи:

- удалить из списка каждый третий элемент;
- переместить минимальный элемент в конец списка;
- отсортировать числа по убыванию;
- определить, есть ли в списке нулевые элементы.

3. Создать множество слов (`HashSet<String>`) из следующего текста: *«Some collection implementations have restrictions on the elements that they may contain. For example, some implementations prohibit null elements, and some have restrictions on the types of their elements. Attempting to add an ineligible element throws an unchecked exception, typically `NullPointerException` or `ClassCastException`. Attempting to query the presence of an ineligible element may throw an exception, or it may simply return false; some implementations will exhibit the former behavior and some will exhibit the latter. More generally, attempting an operation on an ineligible element whose completion would not result in the insertion of an ineligible element into the collection may throw an exception or it may succeed, at the option of the implementation. Such exceptions are marked as optional in the specification for this interface. I need a fish that truly represents my essence. Class `Object` is the root of the class hierarchy. Every class has `Object` as a superclass. All objects, including arrays, implement the methods of this class. The group is running on the beach.»*

Слова разделяются пробелами и запятыми, регистр игнорируется. Решить следующие задачи:

- определите насколько количество элементов множества меньше количества слов в тексте;
- определить можно ли из слов этого множества составить предложение *«collection represents group of objects»*.

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-8: Разрабатывает соответствующие задаче классы для хранения и обработки данных. В классе содержатся

	<p>только необходимые поля. Присутствуют все необходимые методы, все алгоритмы реализованы верно. Каждый метод решает только одну задачу. Среди аргументов методов нет лишних. Выполняет ввод и вывод данных, умеет работать с файлами. Реализует подходящие интерфейсы. Правильно использует коллекции для хранения данных. Правильно решает все задачи.</p> <p>ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса. Использует способы сравнения объектов, определенные стандартом языка Java, использует реализацию стандартных интерфейсов и подходящих стандартных алгоритмов.</p>
Хорошо	<p>ОПК-8: Разрабатывает соответствующие задаче классы для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы, алгоритмы реализованы верно возможно с небольшими неточностями. Каждый метод решает только одну задачу, возможно кроме одного-двух спроектированных недостаточно верно. Выполняет ввод и вывод данных, умеет работать с файлами. Реализует подходящие интерфейсы с небольшими неточностями. Правильно использует коллекции для хранения данных. Правильно решает задачи за исключением небольшого количества (не более одной из каждого из трех разделов задания).</p> <p>ОПК-2: Понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для одного-двух методов неверно назначен уровень доступа. Использует способы сравнения объектов, определенные стандартом языка Java, использует реализацию стандартных интерфейсов и подходящих стандартных алгоритмов, возможно не во всех случаях.</p>
Удовлетворительно	<p>ОПК-8: Разрабатывает соответствующие задаче классы для хранения и обработки данных. В классе содержатся необходимые поля, но могут присутствовать лишние. Присутствуют не все необходимые методы или добавлены явно лишние. Алгоритмы реализации методов содержат ошибки, но не более двух грубых. Некоторые методы спроектированных недостаточно верно. Выполняет ввод и вывод данных с ошибками. Реализует подходящие интерфейсы, но с ошибками. Использует коллекции для хранения данных, возможно с подсказкой преподавателя. Правильно решает не все задачи (но не менее одной из каждого из трех разделов задания).</p> <p>ОПК-2: Не вполне понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для ряда полей и методов неверно назначен уровень доступа. Использует способы сравнения объектов, определенные стандартом языка Java (реализация стандартных интерфейсов и использование стандартных алгоритмов) с ошибками или реализует</p>

	собственные методы вместо использования стандартных.
Неудовлетворительно	ОПК-8: Не знает или плохо знает принципы объектно-ориентированного программирования. Не может разработать соответствующий задаче класс для хранения и обработки данных. Не может создать метод класса, ошибается в алгоритмах. Не умеет выполнять ввод и вывод данных. Не понимает или не знает принципы хранения данных в коллекциях. ОПК-2: Не понимает принципы инкапсуляции и правила определения доступа к элементам класса. Не умеет реализовывать стандартные интерфейсы и использовать стандартные алгоритмы.

Задания по теме №2. «Графические пользовательские приложения»

1. Рассчитайте калорийность куска вареной колбасы по его весу. Известно, что 100 грамм вареной колбасы содержат примерно 301 Ккал. В окне приложения расположите поле для ввода веса продукта, соответствующие пояснения для пользователя (Label или Text), метку для отображения результата и кнопку «Расчет калорийности». После нажатия на кнопку проверьте правильность введенных данных и выведите сообщение о количестве ккал. Если ввод неверный, то сообщите пользователю, где ошибка.

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-2: Знает принципы разработки графических пользовательских приложений с использованием JavaFX. Умеет выбрать и разместить подходящие визуальные элементы сцены окна приложения. Знает параметры настройки этих элементов. ОПК-8: Умеет создавать однооконное графическое приложение в среде разработки. Поясняет код и изменяет его при необходимости. Использует техническую документацию при разработке программы, материалы лекций и другие источники информации.
Хорошо	ОПК-2: Знает принципы разработки графических пользовательских приложений с использованием JavaFX. Умеет выбрать и разместить подходящие визуальные элементы сцены окна приложения с небольшими подсказками преподавателя. Знает отдельные параметры настройки этих элементов. ОПК-8: Умеет создавать однооконное графическое приложение в среде разработки. Поясняет код и изменяет его при необходимости с небольшими неточностями. Иногда использует техническую документацию при разработке программы, в основном пользуется лекционными материалами.
Удовлетворительно	ОПК-2: Знает принципы разработки графических пользовательских приложений с использованием JavaFX. Умеет выбрать и разместить подходящие визуальные элементы сцены окна приложения с подсказками преподавателя. Не умеет изменять параметры настройки элементов сцены или изменяет только некоторые параметры. ОПК-8: Умеет создавать однооконное графическое приложение в среде разработки. С трудом поясняет и изменяет код. Плохо ориентируется в технической документации и в лекционных

	материалах.
Неудовлетворительно	ОПК-2: Не знает принципы разработки графических пользовательских приложений. Не умеет выбрать и разместить подходящие визуальные элементы сцены окна приложения. Не умеет изменять параметры настройки элементов сцены. ОПК-8: Не умеет создавать однооконное графическое приложение в среде разработки или создает неработающий проект. Не может пояснить код и изменить его. Не ориентируется в технической документации и в лекционных материалах.

Задания по теме №6. «Взаимодействие между элементами приложения. События»

1. Расположите на сцене несколько клавиш с буквами и цифрами (не менее 5), например, как на рисунке:

Клавиши можно изобразить с помощью меток. Опишите обработку событий мыши и клавиатуры:

1) при наведении курсора мышью символ на клавише увеличивается в размерах, когда курсор уходит с клавиши, размер символа становится прежним;

2) при нажатии левой кнопки мыши цвет символа на клавише должен стать красным, при нажатии правой – черным, при двойном щелчке – символ меняется на любой другой.

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-2: Знает принципы разработки графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Четко разделяет классы программы в соответствии с этой архитектурой. Умеет построить связи между частями приложения. Знает и умеет использовать обработку событий мыши и клавиатуры. ОПК-8: Умеет создавать графическое приложение в среде разработки. Знает и использует необходимые графические элементы. Поясняет код и изменяет его при необходимости. Использует техническую документацию при разработке программы, материалы лекций и другие источники информации.
Хорошо	ОПК-2: Знает принципы разработки графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Четко разделяет классы программы в соответствии с этой архитектурой. Умеет построить связи между частями приложения, в некоторых случаях с помощью преподавателя. Знает и умеет использовать обработку событий мыши и клавиатуры по крайней мере одним способом. ОПК-8: Умеет создавать графическое приложение в среде разработки. Знает и использует необходимые графические элементы. Поясняет код и изменяет его при необходимости с небольшими неточностями. Иногда использует техническую документацию при разработке программы, в основном пользуется лекционными материалами.

Удовлетворительно	<p>ОПК-2: Знает принципы разработки графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Но нечетко разделяет классы программы в соответствии с этой архитектурой. Не всегда может правильно построить связи между частями приложения. Знает и умеет использовать обработку некоторых событий мыши и клавиатуры.</p> <p>ОПК-8: Умеет создавать многооконное графическое приложение в среде разработки. Знает и использует основные графические элементы, рассмотренные на лекциях. С трудом поясняет и изменяет код. Плохо ориентируется в технической документации и в лекционных материалах.</p>
Неудовлетворительно	<p>ОПК-2: Не знает принципы разработки графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Не может разделить классы программы в соответствии с этой архитектурой. Не может правильно построить связи между частями приложения. Не может реализовать обработку событий мыши и клавиатуры.</p> <p>ОПК-8: Не умеет создавать графическое приложение в среде разработки или создает неработающий проект. Не знает стандарты разработки графических пользовательских приложений. Не может пояснить код и изменить его. Не ориентируется в технической документации и в лекционных материалах.</p>

Список вопросов к зачету:

1. Основные понятия объектно-ориентированного программирования.
 1. Объект, взаимодействие объектов
 2. Инкапсуляция
 3. Наследование
 4. Полиморфизм
2. Синтаксис языка Java.
 1. Переменные, литералы.
 2. Типы данных.
 3. Операторы.
 4. Описание класса.
 5. Описание полей и методов.
 6. Подключение и использование пакетов.
 7. Массивы.
 8. Методы, передача данных в методы
3. Классы
 1. Поля и методы
 2. Доступ к членам класса.
 3. Объекты, объектные ссылки.
 4. Работа с объектами: создание, вызов методов, сравнение.
4. Наследование.
 1. Правила создания и использования классов-потомков.
 2. Доступ к членам класса-предка.
 3. Наследование методов, переопределение и перегрузка.

4. Интерфейсы. Реализация интерфейсов
5. Абстрактные классы
5. Обработка исключительных ситуаций
 1. Генерация исключений
 2. Блок перехвата исключений и блок-ловушка
 3. Стандартные исключения.
 4. Пользовательские исключения.
6. Ввод/вывод.
 1. Потокковые классы для ввода/вывода.
 2. Работа с файлами.
 3. Класс Scanner.
7. Строки.
 1. Обработка текстовой информации с использованием класса String.
 2. Основные методы класса, операции со строками.
8. Классы-обертки, обертки примитивных типов
9. Контейнеры.
 1. Виды контейнеров.
 2. Свойства контейнеров.
 3. Хранение объектов библиотечных классов и пользовательских.
 4. Итераторы.
 5. Алгоритмы.
10. Основные элементы графических пользовательских приложений JavaFX.
 1. Сцена
 2. Элементы отображения информации
 3. Управляющие элементы
 4. Обработка события нажатия кнопки
 5. Расположение элементов на сцене
 6. Однооконное приложение
11. Архитектура графического пользовательского приложения Модель-вид-контроллер
 1. Разработка модели.
 2. Активная и пассивная модель
 3. Отображение данных.
 4. Взаимодействие классов «модель» и «вид»
 5. Функции контроллера
12. Диалоги.
 1. Разработка диалога для ввода данных.
 2. Проверка вводимых данных
 3. Стандартные диалоги
13. Меню, контекстное меню
14. Обработка событий
 1. События мыши
 2. События клавиатуры

Приложение № 2 к рабочей программе дисциплины «Информационные системы и технологии»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Информационные системы и технологии» в первой половине курса являются лекции, так как изучение языка программирования требует знания определенного количества теоретического материала, причем в достаточно большом объеме. Во второй половине курса большинство занятий представляют из себя лекции-беседы, на которых обсуждаются технологии программирования и конкретные примеры их реализации, а так же мастер-классы, где преподаватель демонстрирует процесс разработки программного обеспечения, основные его аспекты и обсуждает возможности рассматриваемых инструментов разработки.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. В основном такими задачами являются лабораторные работы различного объема, а так же небольшие задачи связанные с исправлением ошибок и доработкой программ. Примеры решения задач разбираются на лекциях, кроме того проводятся лабораторные занятия. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной на консультациях, практических занятиях или из учебной литературы.

Для усвоения материала необходимо в течение всего курса выполнять задания для самостоятельной работы.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков разработки программ, в течение обучения проводятся мероприятия текущей аттестации в виде письменных контрольных работ в обоих семестрах изучения дисциплины. Также проводятся консультации (при необходимости) по разбору заданий, которые вызвали затруднения.

В конце изучения дисциплины студенты сдают экзамен. Экзамен принимается в письменном виде и включает в себя, как теоретические вопросы, так и практические, связанные с написанием программ. На самостоятельную подготовку к экзамену выделяется 3 дня, во время подготовки к экзамену предусмотрена групповая консультация.

Освоить вопросы, излагаемые в процессе изучения дисциплины «Информационные системы и технологии» самостоятельно студенту довольно сложно. Это связано со сложностью изучаемого материала и большим объемом курса. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра сдать зачет и экзамен по итогам изучения дисциплины студенту практически невозможно.