

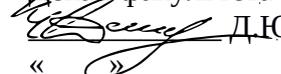
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

«          » 2020 г.

**Рабочая программа дисциплины**  
«Языки и методы программирования»

**Направление подготовки**  
09.03.03 Прикладная информатика

Направленность (профиль) «Прикладная информатика в экономике»

**Форма обучения**  
очная

Программа рассмотрена  
на заседании кафедры  
от 31 августа 2020 г.,  
протокол № 1

Зав. кафедрой

  
В.В. Васильчиков

Программа одобрена НМК  
факультета ИВТ  
протокол № 1 от  
от 31 августа 2020 г.

Председатель НМК

  
Г.В. Шабаршина

Ярославль  
2020

### 1. Цели освоения дисциплины

Целями дисциплины «Языки и методы программирования» 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 ОП бакалавриата

Дисциплина «Языки и методы программирования» относится к вариативной части ОП бакалавриата.

Для освоения данной дисциплиной студенты должны обладать знаниями по математике и информатике в объеме школьной программы, а так же в объеме курса «Основы программирования», проявлять настойчивость, целеустремленность и инициативу в процессе обучения.

Знания и навыки, полученные при ее изучении, используются учащимися при изучении последующих дисциплин профессионального цикла, а именно «Программная инженерия», «Современные мобильные платформы и сервисы», а также для продолжения обучения в магистратуре по направлению Прикладная математика и информатика. Материал лекций является основой для двух компьютерных практикумов: «Практикум ЭВМ по объектно-ориентированному программированию» и «Практикум ЭВМ по языкам программирования». Лабораторные занятия этих дисциплин предназначены для практического усвоения знаний и умений, получаемых на лекциях дисциплины «Языки и методы программирования».

### 3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

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

Код компетенции	Формулировка компетенции	Перечень планируемых результатов обучения
<b>Общепрофессиональные компетенции</b>		
ПК-2	способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение	<b>Знать:</b> <ul style="list-style-type: none"><li>• основы теории алгоритмов и ее применения;</li><li>• синтаксис языка программирования Java;</li><li>• принципы объектно-ориентированного программирования.</li></ul> <b>Уметь:</b> <ul style="list-style-type: none"><li>• разрабатывать программы на языке Java;</li><li>• создавать графические пользовательские приложения с использованием JavaFX;</li><li>• выполнять отладку и тестирование программы;</li><li>• разрабатывать приложения в рамках архитектуры «Модель-</li></ul>

		<p>Вид-Контроллер».</p> <p>Владеть навыками:</p> <ul style="list-style-type: none"> <li>• работы с интегрированной средой разработки;</li> <li>• работы с научно-технической литературой и технической документацией Java и JavaFX.</li> </ul>
--	--	--

#### 4. Объем, форма реализации, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зач. ед., 180 акад. Час.

Дисциплина реализуется с применением дистанционных образовательных технологий (ДОТ), предоставляемых образовательной площадкой MOOK ЯрГУ им. П.Г. Демидова (DemidOnline).

Отдельные элементы курса преподаются в дистанционной форме в рамках онлайн курса «Разработка графических приложений на JavaFX», размещённого по ссылке: <https://demidonline.uni-yar.ac.ru/courses/course-v1:DemidOnline+JApp001x+2020/about>

№ 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	
			Контактная работа						Форма промежуточной аттестации (по семестрам)	
			лекции	практические	лабораторные	консультации	аттестационные испытания	работасамостоятельная		
1	Синтаксис языка Java	7	2		2			4	Лабораторная работа	
2	Объектно-ориентированное программирование.	7	2		2	1		4	Лабораторная работа	
3	Контейнеры, коллекции	7	2		2	1		6	Контрольная работа	
4	Основные понятия JavaFX	7	1		4			4		
5	Компоненты JavaFX	7	1		4			4	Лабораторная работа	
6	Элементы архитектуры Модель-Вид-Контроллер и их взаимосвязь	7	1		4	1		4	Лабораторная работа	
7	Компоненты-свойства (property) JavaFX	7	1		4			4		
8	Диалоговые окна	7	1		4	1		4	Лабораторная работа	
9	Главное меню приложения	7	1		4			4		

	и другие виды меню								
10	Обработка событий	7	1		4			4	Лабораторная работа
11	Компоненты отображения наборов данных: таблицы, списки, деревья	7	1		4	1		4	
12	Редактирование данных: добавление, изменение, удаление	7	1		4			6	Контрольная работа
13	Работа со списком объектов. Выполнение подзадач в отдельных потоках	7	1		4	1		4	
14	Основные элементы и технологии FXML	7	1		4	1		4	Лабораторная работа
15	Оформление внешнего вида приложения с помощью таблиц стилей	7	1		4	1		4	
	Экзамен	7				2	0,5	33,5	Экзамен
	<b>Всего</b>		<b>18</b>		<b>54</b>	<b>10</b>	<b>0,5</b>	<b>97,5</b>	

### Содержание разделов дисциплины:

1. *Очно или дистанционно в рамках собственной внутренней образовательной среды LMS Moodle ЯрГУ.* Синтаксис языка Java. Базовые конструкции языка Java. Переменные. Типы данных, преобразование типов.

2. *Очно или дистанционно в рамках собственной внутренней образовательной среды LMS Moodle ЯрГУ.* Объектно-ориентированное программирование. Основные понятия. Объект, класс, метод. Инкапсуляция. Наследование. Полиморфизм. Классы. Элементы класса. Поля и методы. Конструкторы. Наследование и полиморфизм. Базовые и производные классы. Доступ к элементам класса. Переопределение методов. Абстрактные классы. Интерфейсы. Определение интерфейса. Реализация интерфейса. Стандартные интерфейсы.

3. *Очно или дистанционно в рамках собственной внутренней образовательной среды LMS Moodle ЯрГУ.* Контейнеры, коллекции. Основные классы и интерфейсы. Итераторы. Последовательности и множества. Свойства и преимущества контейнеров.

4. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Основные понятия JavaFX. Структура графического приложения. Основные классы структуры: Application, Stage, Scene. Пример однооконного программного приложения JavaFX в среде разработки IntelliJ IDEA.

5. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Компоненты JavaFX. Визуальные компоненты библиотеки JavaFX: Label, Text, TextField, Button и др. Настройка параметров компонентов. Управление расположением компонентов: GridPane, VBox, HBox, TabPane и др.. Сложные компоненты для построения графиков и диаграмм: BarChart, LineChart, PieChart. Примеры программных приложений с использованием рассматриваемых компонентов.

6. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Элементы архитектуры Модель-Вид-Контроллер и их взаимосвязь. Основные элементы архитектуры Модель-Вид-Контроллер. Назначение и особенности программной реализации модели, вида и контроллера. Создание взаимосвязи между элементами архитектуры. Шаблон Наблюдатель (Слушатель) как пример архитектуры Модель-Вид-Контроллер. Пример программного приложения в рамках архитектуры Модель-Вид-Контроллер.

7. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Компоненты-свойства (property) JavaFX.

Компоненты пакета `javafx.beans.property` как пример классов, реализующих шаблон Наблюдатель (Слушатель). Соглашение об именах и использование компонентов-свойств в программных приложениях. Поля-свойства компонентов JavaFX. Создание взаимосвязи между свойствами. Пример программного приложения с использованием компонентов-свойств.

8. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Диалоговые окна. Стандартные диалоговые окна. Окна сообщения, ввода текста, выбора из списка. Диалог открытия файлов. Примеры использования компонентов JavaFX для создания стандартных диалоговых окон. Пользовательские диалоговые окна. Проверка вводимых данных. Пример программного приложения с пользовательским диалоговым окном.

9. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Главное меню приложения и другие виды меню. Компоненты JavaFX для создания разных видов меню. Главное меню приложения `MenuBar`, отдельные меню и подменю `Menu`, пункты меню `MenuItem`, контекстное меню `ContextMenu`, кнопочное меню `SplitMenuButton`. Пример программного приложения с использованием разных видов меню.

10. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Обработка событий. События. Типы событий. Получение информации о событиях. Технология обработки и перехвата событий в JavaFX. Обработчики событий. Пример программного приложения с обработкой событий мыши и клавиатуры.

11. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Компоненты отображения наборов данных: таблицы, списки, деревья. Отображение данных в виде таблицы `TableView`. Пример программного приложения для хранения и редактирования списка объектов с отображением их в виде таблицы. Отображение данных в виде дерева `TreeView`. Пример программного приложения для отображения данных иерархической структуры в виде дерева.

12. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Редактирование данных: добавление, изменение, удаление. Взаимодействие компонентов графического приложения в процессе добавления, изменения и удаления данных. Интеграция в одном графическом приложении хранения, отображения данных, использования диалоговых окон, пользовательского меню. Пример программного приложения, интегрирующего технологии отображения данных и взаимодействия с пользователем.

13. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Работа со списком объектов. Выполнение подзадач в отдельных потоках. Отображение данных в виде списка `ListView`. Пример приложения для отбора и отображения списка объектов. Создание отдельных потоков в графических приложениях JavaFX с помощью класса `Task`.

14. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Основные элементы и технологии FXML. Правила описания пользовательского интерфейса с использованием технологии FXML и создание fxml-документа. Способ интеграции fxml-документа в графическое приложение. Пример программного приложения JavaFX с технологией FXML в среде разработки IntelliJ IDEA.

15. *Дистанционно на площадке DemidOnline в рамках курса «Разработка графических приложений на JavaFX».* Оформление внешнего вида приложения с помощью таблиц стилей. Правила описания стилей для технологии FXML. Создание таблиц стилей для графического приложения на JavaFX. Интеграция таблиц стилей в приложение. Пример добавления таблицы стилей в программное приложение.

## **5. 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.

## **6. Перечень электронных ресурсов и информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)**

В процессе осуществления образовательного процесса используются:

– материалы онлайн курса «Разработка графических приложений на JavaFX», размещённого на образовательной онлайн площадке ЯрГУ им. П.Г. Демидова DemidOnline по ссылке:

<https://demidonline.uniyar.ac.ru/courses/course-v1:DemidOnline+JApp001x+2020/about>  
для разработки документов, презентаций, для работы с электронными таблицами  
OfficeStd 2013 RUS OLP NL Acdmc 021-10232

LibreOffice (свободное)

издательская система LaTeX;

– для выполнения лабораторных работ

среда разработки IntelliJ IDEA (свободная)

## **7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

а) основная:

1. Лагутина, Н. С., Основы объектно-ориентированного программирования на языке JAVA: учебно-методическое пособие / Н. С. Лагутина, Ю. А. Ларина; Яросл. гос. ун-т им. П. Г. Демидова. - Ярославль: ЯрГУ, 2019. - 79 с.

2. Сьерра, Кэти. Изучаем Java: пер. с англ / К. Сьерра, Б. Бэйтс. - 2-е изд. - М.: Эксмо, 2020. - 717 с.: ил. - (Мировой компьютерный бестселлер).

б) дополнительная:

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.  
<http://www.lib.uniyar.ac.ru/edocs/iuni/20140402.pdf>

4. Гонсалвес, Э., Изучаем Java EE 7 / Э. Гонсалвес; [пер. с англ. Е. Зазноба и др.], СПб., Питер, 2016, 640с

5. Эккель, Б., Философия Java / Б. Эккель; пер. с англ. Е. Матвеева. - 4-е полное изд., М., Питер, 2017, 1165с

в) ресурсы сети «Интернет»

– Среда разработки IntelliJ IDEA: <https://www.jetbrains.com/ru-ru/idea/> Доступ свободный

– Документация java 8: <https://docs.oracle.com/javase/8/docs/api/>. Доступ свободный

– Документация javaFX: <https://docs.oracle.com/javase/8/javafx/api/toc.htm>. Доступ свободный

– материалы онлайн курса «Разработка графических приложений на JavaFX», размещённого на образовательной онлайн площадке ЯрГУ им. П.Г. Демидова DemidOnline по ссылке:

<https://demidonline.uniyar.ac.ru/courses/course-v1:DemidOnline+JApp001x+2020/about>

– Электронная библиотека учебных материалов ЯрГУ

([http://www.lib.uniyar.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php)).

– Электронно-библиотечная система «Университетская библиотека online» ([www.biblioclub.ru](http://www.biblioclub.ru)).

## **8. 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 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.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

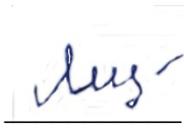
Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в

электронную информационно-образовательную среду организации, а также материалам онлайн курса «Разработка графических приложений на JavaFX», размещённого на образовательной онлайн площадке ЯрГУ им. П.Г. Демидова (DemidOnline).

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

**Автор(ы) :**

Доцент кафедры ВПС, к.ф.-м.н.



Н.С.Лагутина

**Приложение №1 к рабочей программе дисциплины  
«Языки и методы программирования»  
Фонд оценочных средств  
для проведения текущей и промежуточной аттестации студентов  
по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

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

**Задания для лабораторных работ**

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 78 13 2 5 78 2 5 6	2
2 1 2	2
6	6

Критерии оценивания

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

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

Примеры заданий по теме «Объектно-ориентированное программирование»:

1. 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.

2. 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	ПК-2: 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. 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	ПК-2: 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

	<p>верно. Умеет создавать объекты классов и работать с ними.</p> <p>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</p>
Удовлетворительно	<p>ПК-2: 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.</p> <p>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.</p>
Неудовлетворительно	<p>ПК-2: 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.</p> <p>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.</p>

#### Примеры заданий по теме «Компоненты JavaFX»

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>ПК-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. Умеет работать с текстовыми и графическими файлами как с ресурсами проекта. Знает и использует способы изменения внешнего вида элементов графического интерфейса.</p> <p>Умеет создавать однооконное графическое приложение в среде</p>

	разработки. Поясняет код и изменяет его при необходимости. Использует техническую документацию при разработке программы, материалы лекций и другие источники информации.
Хорошо	ПК-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. Умеет работать с текстовыми и графическими файлами как с ресурсами проекта. Знает и использует некоторые способы изменения внешнего вида элементов графического интерфейса. 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	ПК-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. Умеет работать с текстовыми файлами. Не знает или использует только некоторые способы изменения внешнего вида элементов графического интерфейса. Умеет создавать однооконное графическое приложение в среде разработки. С трудом поясняет и изменяет код. Плохо ориентируется в технической документации и в лекционных материалах.
Неудовлетворительно	ПК-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. Не знает способы изменения внешнего вида элементов графического интерфейса. 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.

Примеры заданий по теме «Элементы архитектуры Модель-Вид-Контроллер и их взаимосвязь»:

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	<p>ПК-2: 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.</p> <p>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. Использует техническую документацию при разработке программы, материалы лекций и другие источники информации.</p>
Хорошо	<p>ПК-2: 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</p>

	<p>программы.</p> <p>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>ПК-2: 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>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>ПК-2: 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>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>

#### Задания по теме «Обработка событий»

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	ПК-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. 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	ПК-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. 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	ПК-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. 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.
Неудовлетворительно	ПК-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. 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.

Задания по теме «Диалоговые окна»

1. Абитуриент поступает в вуз. В вузе есть три направления, на каждое из которых известен проходной балл: Прикладной мыслитель (210), Фундаментальный лентяй (234), Кабалистика и ворожба (203). Абитуриент приносит паспорт с именем и баллы ЕГЭ по трем предметам: матендра, пыхтелка, очумелые ручки; по каждому предмету баллы от 0 до 100. Абитуриент проходит на направление, если сумма баллов по трем предметам равна или превышает соответствующий проходной балл. В главном окне приложения отобразите имя абитуриента и сообщение о том на какие направления он может пройти (можно добавить информацию о баллах). Так же на главной сцене разместите кнопку для вызова диалогового окна для редактирования данных об абитуриенте. Соответственно разработайте диалоговое окно для редактирования имени абитуриента и баллов по трем предметам. Проверка вводимых данных приветствуется.

#### Критерии оценивания

Оценка	Критерии
Отлично	ПК-2: 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. Знает и использует стандартные и пользовательские диалоги. Создает подходящие классы и объекты. Умеет создавать многооконное графическое приложение в среде разработки. Знает и использует необходимые графические элементы, в том числе соответствующие стандартам разработки пользовательских приложений. Поясняет код и изменяет его при необходимости. Использует техническую документацию при разработке программы, материалы лекций и другие источники информации.
Хорошо	ПК-2: 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. Знает и использует стандартные и пользовательские диалоги. Создает подходящие классы и объекты с небольшими ошибками. 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.
Удовлетворительно	ПК-2: 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. Знает и использует основные стандартные диалоги. Создает подходящие классы и объекты с ошибками. 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.
Неудовлетворительно	ПК-2: 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 создать подходящий стандартный диалог. Не знает принципы построения пользовательского диалога. 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.

Задания по теме «Основные элементы и технологии FXML»

1. Создайте FXML-приложение окно которого имитирует внешний вид устройства, изображенного на рисунке (это не существующее устройство было придумано дизайнером с ностальгией по советским временам). Окно приложения не обязательно должно в точности повторять все детали рисунка, например цвета, но чем точнее, тем лучше.

Дополнительно:

\* Реализуйте обработку кнопок с цифрами так, чтобы в центральном поле отображалась нажатая цифра

\*\* Реализуйте обработку кнопок с цифрами так, чтобы в центральном поле отображались все нажатые цифры

Критерии оценивания

Оценка	Критерии
Отлично	ПК-2: Знает принципы разработки графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Четко разделяет классы программы в соответствии с этой архитектурой. Знает и использует принципы создания декларативного пользовательского интерфейса с использованием FXML. Реализует алгоритмы выполнения функций приложения в соответствии с поставленной задачей в том числе дополнительные. Умеет создавать графическое приложение в среде разработки. Знает и использует необходимые графические элементы в соответствии с

	<p>технологией FXML, в том числе 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>ПК-2: Знает принципы разработки графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Четко разделяет классы программы в соответствии с этой архитектурой. Знает и использует принципы создания декларативного пользовательского интерфейса с использованием FXML с помощью преподавателя. Реализует алгоритмы выполнения основных функций приложения в соответствии с поставленной задачей. Умеет создавать графическое приложение в среде разработки. Знает и использует необходимые графические элементы в соответствии с технологией FXML, 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>ПК-2: Знает принципы разработки графических пользовательских приложений в рамках архитектуры «Модель-Вид-Контроллер». Но нечетко разделяет классы программы в соответствии с этой архитектурой. Знает и использует принципы создания декларативного пользовательского интерфейса с использованием FXML в значительной части с помощью преподавателя. Реализует алгоритмы выполнения основных функций приложения в соответствии с поставленной задачей с небольшими неточностями. Умеет создавать графическое приложение в среде разработки. Знает и использует необходимые графические элементы в соответствии с технологией FXML, рассмотренные на лекциях. С трудом поясняет и изменяет код. Плохо ориентируется в технической документации и в лекционных материалах.</p>
Неудовлетворительно	<p>ПК-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 знает принципы создания декларативного пользовательского интерфейса с использованием FXML. Не может реализовать функции приложения или реализует с существенными ошибками. Не умеет создавать графическое приложение в среде разработки или создает неработающий проект. Не знает стандарты разработки графических пользовательских приложений. Не может</p>

	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).

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	ПК-2: Разрабатывает соответствующий задаче класс для хранения и обработки данных. В классе содержатся только необходимые поля. Присутствуют все необходимые методы, все алгоритмы реализованы верно.

	<p>Каждый метод решает только одну задачу. Среди аргументов методов нет лишних. Выполняет ввод и вывод данных, умеет работать с файлами. Реализует подходящие интерфейсы.</p> <p>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).</p>
Хорошо	<p>ПК-2: 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.</p> <p>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.</p>
Удовлетворительно	<p>ПК-2: 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.</p> <p>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.</p>
Неудовлетворительно	<p>ПК-2: 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.</p> <p>Не понимает принципы инкапсуляции и правила определения доступа к элементам класса. Не умеет реализовывать стандартные интерфейсы и использовать стандартные алгоритмы.</p>

## Контрольная работа 2

### 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 OX: A, B, C. Определить, какая из точек B или C расположена ближе к A.

### 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	ПК-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. Реализует алгоритмы выполнения функций приложения в соответствии с поставленной задачей. Знает и использует необходимые графические элементы, в том числе соответствующие стандартам разработки пользовательских приложений: элементы отображения и ввода информации.
Хорошо	ПК-2: 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. Знает и использует необходимые графические элементы, в том числе соответствующие стандартам разработки

	пользовательских приложений: элементы отображения и ввода информации, но допускает синтаксические ошибки.
Удовлетворительно	ПК-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. Реализует большую часть алгоритмов выполнения функций приложения в соответствии с поставленной задачей, но не все или с существенными ошибками. Знает и использует основные графические элементы, рассмотренные на лекциях. Допускает много синтаксических ошибок, некоторые элементы использует некорректно.
Неудовлетворительно	ПК-2: 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 стандарты разработки графических пользовательских приложений. Не знает синтаксис элементов графического приложения и их методов.

#### Список вопросов к экзамену:

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. Поточковые 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

### Пример экзаменационной работы

1. Найдите ошибки

```
interface Shape {
    double Square();
    double Perimeter();
}
class Triangle implements Shape {
    private double a, b;
    double P () {
        return (a+b+c)/2;
    }
}
```

```

    public double Perimeter(){return (a+b+c)};
}

```

2. Что такое контейнеры. Опишите общие свойства контейнеров-коллекций.
3. Опишите правила обработки событий в JavaFX. Приведите пример обработки события мыши или клавиатуры.
4. Написать консольное приложение для решения задачи: вводится целое число в десятичной системе счисления, вывести это число с разделением на триады цифр.
5. Создать графическое приложение с использованием библиотеки JavaFX, в котором можно ввести номер группы студента нашего факультета и получить сообщение о корректности ввода.

Показатели и критерии, используемые при выставлении оценки:

Номер задачи	Критерии	Шкала оценивания
1	ПК-2: Знать: <ul style="list-style-type: none"> <li>• синтаксис языка программирования Java;</li> <li>• принципы объектно-ориентированного программирования.</li> </ul> Уметь: <ul style="list-style-type: none"> <li>• выполнять отладку и тестирование программы.</li> </ul>	0 баллов – студент полностью не верно решил задачу (в приведенном примере задания не нашел ни одной ошибки верно); 1 балл – студент частично разобрался в примере кода и решил поставленную задачу с ошибкой (в приведенном примере задания нашел одну ошибку или нашел лишние ошибки) 2 балла – студент полностью разобрался в примере кода и правильно решил поставленную задачу (в приведенном примере задания нашел две ошибки)
2	ПК-2: Знать: <ul style="list-style-type: none"> <li>• синтаксис языка программирования Java;</li> <li>• принципы объектно-ориентированного программирования.</li> </ul> Владеть навыками: <ul style="list-style-type: none"> <li>• работы с научно-технической литературой и технической документацией Java и JavaFX.</li> </ul>	0 баллов – студент не знает определение и принципы работы элементов языка программирования Java; 1 балл – студент в основном описал и проанализировал элементы языка программирования с учетом технологий программирования (в приведенном примере задания привел и описал свойства большей части типов коллекций: последовательных – списки и динамические массивы, множеств и упорядоченных множеств), возможно с небольшими неточностями или не полностью; 2 балла – студент полностью описал и проанализировал элементы языка программирования с учетом технологий программирования (в приведенном примере задания привел и описал свойства всех типов коллекций: последовательных – списки и динамические массивы, множеств и

		упорядоченных множеств)
3	<p>ПК-2:</p> <p>Знать:</p> <ul style="list-style-type: none"> <li>• синтаксис языка программирования Java и библиотеки JavaFX.</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>• использовать методы защиты данных.</li> </ul>	<p>0 баллов – студент не знает определение и принципы работы элементов языка программирования Java и библиотеки JavaFX;</p> <p>1 балл – студент в основном описал и проанализировал элементы библиотеки JavaFX с учетом технологий программирования и методов защиты данных, возможно с небольшими неточностями или не полностью;</p> <p>2 балла – студент полностью описал и проанализировал элементы библиотеки JavaFX с учетом технологий программирования и методов защиты данных</p>
4	<p>ПК-2:</p> <p>Знать:</p> <ul style="list-style-type: none"> <li>• основы теории алгоритмов и ее применения;</li> <li>• синтаксис языка программирования Java;</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>• разрабатывать программы на языке Java;</li> </ul>	<p>0 баллов – студент не может написать полноценное консольное приложение на языке Java или не верно разработана модель решения задачи;</p> <p>1 балл – студент в основном может написать полноценное консольное приложение на языке Java и разработать и реализовать модель и алгоритмы решения поставленной задачи, но присутствуют ошибки в построенной модели, в реализации алгоритмов или существенные ошибки в применении средств языка</p> <p>2 балла – студент может написать полноценное консольное приложение на языке Java и правильно разработать и реализовать модель и алгоритмы решения поставленной задачи с небольшими ошибками в реализации отдельных частей модели и алгоритмов или ошибками в применении средств языка</p> <p>3 балла – студент может написать полноценное консольное приложение на языке Java и правильно разработать и реализовать модель и алгоритмы решения поставленной задачи</p>
5	<p>ПК-2:</p> <p>Знать:</p> <ul style="list-style-type: none"> <li>• разрабатывать приложения в рамках архитектуры «Модель-Вид-Контроллер».</li> <li>• принципы объектно-ориентированного программирования.</li> </ul>	<p>0 баллов – студент не может написать полноценное графическое приложение на языке Java с использованием библиотеки JavaFX или не верно реализованы алгоритмы решения задачи;</p> <p>1 балл – студент в основном может написать полноценное графическое приложение на языке Java с использованием библиотеки JavaFX, но</p>

	<p>Уметь:</p> <ul style="list-style-type: none"> <li>создавать графические пользовательские приложения с использованием JavaFX;</li> </ul>	<p>присутствуют ошибки, в реализации алгоритмов или существенные ошибки в применении средств языка программирования</p> <p>2 балла – студент может написать полноценное графическое приложение на языке Java с использованием библиотеки JavaFX и правильно разработать и реализовать модель и алгоритмы решения поставленной задачи в соответствии с архитектурой «Модель-Вид-Контроллер»</p>
	<p>ПК-2:</p> <p>Уметь:</p> <ul style="list-style-type: none"> <li>использовать методы защиты данных;</li> </ul>	<p>0 баллов – модель решения задачи построена не верно;</p> <p>1 балл – правильно построена модель решения задачи с учетом защиты данных и организации связей между элементами программы</p>

Максимальное количество баллов по ПК-2 – 12 баллов.

Набранное количество баллов соответствует оценке за выполнение экзаменационной работы:

- менее 5 баллов — оценка «неудовлетворительно»,
- от 5 до 7 баллов — оценка «удовлетворительно», пороговый уровень формирования компетенции,
- от 8 до 10 баллов — оценка «хорошо», продвинутый уровень формирования компетенции,
- от 11 до 12 баллов — оценка «отлично», высокий уровень формирования компетенции.

#### **Методические указания по выставлению итоговой оценки за экзамен.**

Для успешного освоения дисциплины обязательно:

- прохождение онлайн курса «Разработка графических приложений на JavaFX» на площадке MOOK ЯрГУ им. П.Г. Демидова (DemidOnline) (является допуском к прохождению процедуры промежуточной аттестации);
- выполнение лабораторных и контрольных работ (являются формой текущей аттестации).

Итоговая оценка по дисциплине «Языки и методы программирования» формируется в два этапа.

Первый этап – оценивание работы студента в течение изучения курса на основе средней оценки за лабораторные и контрольные работы (всего семь мероприятий текущей аттестации). Если на этом этапе все аттестационные задания выполнены в срок и средний балл за текущую аттестацию больше трех студенту может быть выставлена итоговая оценка «удовлетворительно» автоматом досрочно. Если все аттестационные задания выполнены в срок и средний балл за текущую аттестацию больше четырех студенту может быть выставлена итоговая оценка «хорошо» автоматом досрочно. Если все аттестационные задания выполнены в срок и средний балл за текущую аттестацию равен пяти студенту может быть выставлена итоговая оценка «отлично» автоматом досрочно.

Второй этап – проведение экзаменационной работы. Для получения положительной оценки за экзамен студент должен выполнить все лабораторные работы на положительную оценку, возможно не полностью в срок. При выполнении этого условия оценка за экзаменационную работу считается итоговой.