

МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 23 » мая 2023 г.

Рабочая программа дисциплины
«Языки и методы программирования»

Направление подготовки
01.03.02 Прикладная математика и информатика

Направленность (профиль)
«Программирование и технологии искусственного интеллекта»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа рассмотрена на
заседании кафедры
от 21 апреля 2023 г.,
протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 6 от
28 апреля 2023 г.

Ярославль

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 базовой 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:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		

		<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.
<p>ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</p>	<p>ОПК – 5.1 Демонстрирует знания методов алгоритмизации, языков и технологий программирования, пригодных для практического применения.</p> <p>ОПК – 5.2 Демонстрирует умение применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач</p> <p>ОПК – 5.3 Демонстрирует навыки: программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>	<p>Знать:</p> <ul style="list-style-type: none"> • синтаксис языка программирования Java и библиотеки JavaFX; • принципы объектно-ориентированного программирования. <p>Уметь:</p> <ul style="list-style-type: none"> • разрабатывать программы на языке Java; • использовать методы защиты данных; • выполнять отладку программы. <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 5 зач. ед., 180 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость					Формы текущего контроля успеваемости Форма промежуто	
			Контактная работа	лекции	практические	лабораторные консультации	аттестационные испытания		самостоятельная работа
1.	Объектно- ориентированное программирование. Основные понятия.	3	4					1	
2.	Синтаксис языка Java	3	4			1		4	Задания для самостоятельной работы
3.	Классы. Элементы класса.	3	4					4	Контрольная работа №1
4.	Управление доступом. Инкапсуляция.	3	4					4	
5.	Наследование и полиморфизм	3	4			1		4	Задания для самостоятельной работы
6.	Интерфейсы	3	4					4	
7.	Обработка исключительных ситуаций.	3	4					4	
8.	Основные классы для ввода/вывода.	3	4					4	
9.	Обработка текстовой информации с использованием класса String.	3	4			1		4	Задания для самостоятельной работы
	Всего за 3 семестр		36			3		33	
10.	Контейнеры, коллекции	4	4			1		4	Контрольная работа №2
11.	Графические пользовательские приложения	4	4					2	
12.	Введение в JavaFX	4	4					2	
13.	Архитектура создания пользовательских приложений: Модель Вид Контроллер.	4	4			1		4	
14.	Основные элементы сцены	4	4			1		4	Задания для самостоятельной работы
15.	Взаимодействие между элементами приложения. События	4	4					4	

.									
16	Организация ввода данных. Диалоговые окна.	4	4					4	
17	Главное окно приложения.	4	4			1		4	Задания для самостоятельной работы
18	Разработка декларативного интерфейса с помощью FXML	4	4					4	
	Всего за 4 семестр		36			4		32	Экзамен
	Всего		72			7	36	65	

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, не визуальные 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, 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 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: 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 NetBeans 8 (свободная)

– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

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

а) основная:

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

5. Лагутина Н. С. Основы объектно-ориентированного программирования на языке JAVA [Электронный ресурс]: учебно-методическое пособие. / Н. С. Лагутина, Ю. А. Ларина; Яросл. гос. ун-т им. П. Г. Демидова - Ярославль: ЯрГУ, 2019. - 79 с. <http://www.lib.uniyar.ac.ru/edocs/iuni/20190403.pdf>

б) дополнительная:

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. <http://www.lib.uniyar.ac.ru/edocs/iuni/20060480.pdf>

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

в) ресурсы сети «Интернет»

– Среда разработки NetBeans 8.0.3: www.netbeans.org. Доступ свободный

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

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

- Электронно-библиотечная система «Юрайт»(<https://urait.ru/>).

- Электронно-библиотечная система «Лань»(<https://e.lanbook.com/>).

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.

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

Критерии оценивания

Оценка	Критерии
Отлично	ОПК-3: Знает и применяет подходящие для решения алгоритмы, выбирает наиболее эффективный алгоритм. Знает и применяет синтаксические конструкции языка Java. Программа выполняется для всех тестов ОПК-4: Создает полноценное консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости
Хорошо	ОПК-3: 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-4: Создает консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости с небольшими

	неточностями
Удовлетворительно	ОПК-3: Знает и применяет простые алгоритмы. Знает и применяет синтаксические конструкции языка Java. Программа выполняется для всех тестов, возможно кроме одного-двух крайних случаев. ОПК-4: Создает консольное приложение в среде разработки. С трудом поясняет код, не может изменить код при усложнении или существенном дополнении задачи
Неудовлетворительно	ОПК-3: Не может подобрать подходящие для решения алгоритмы. Ошибается во многих синтаксических конструкциях языка Java. Программа не выполняется для большинства тестов. ОПК-4: 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»:

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	ОПК-3: 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-4: 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-3: 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-4: 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-3: 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-4: 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-3: 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-4: 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-3: Знает класс String и применяет методы этого класса, подходящие для решения задачи. Знает и применяет регулярные выражения. Программа выполняется для всех тестов ОПК-4: Создает полноценное консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости
Хорошо	ОПК-3: 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-4: Создает консольное приложение в среде разработки. Поясняет код и изменяет его при необходимости с небольшими неточностями
Удовлетворительно	ОПК-3: 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-4: Создает консольное приложение в среде разработки. С трудом поясняет код, не может изменить код при усложнении или существенном дополнении задачи
Неудовлетворительно	ОПК-3: 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-4: 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-3: 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-4: 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-3: 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-4: 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-3: 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-4: 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-3: 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-4: 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.</p>

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-3: 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-4: 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.

	<p>Поясняет код и изменяет его при необходимости. Использует техническую документацию при разработке программы, материалы лекций и другие источники информации.</p>
Хорошо	<p>ОПК-3: 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-4: 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-3: 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-4: 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-3: 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-4: Не умеет создавать графическое приложение в среде разработки или создает неработающий проект. Не знает стандарты разработки графических пользовательских приложений. Не может пояснить код и</p>

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-3: 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-4: Понимает принципы инкапсуляции и правила определения доступа к элементам класса.
Хорошо	ОПК-3: 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-4: Понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для одного-двух методов неверно назначен уровень доступа.
Удовлетворительно	ОПК-3: Разрабатывает соответствующий задаче класс

	<p>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.</p> <p>ОПК-4: Не вполне понимает принципы инкапсуляции и правила определения доступа к элементам класса, возможно для ряда полей и методов неверно назначен уровень доступа.</p>
Неудовлетворительно	<p>ОПК-3: 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.</p> <p>ОПК-4: Не понимает принципы инкапсуляции и правила определения доступа к элементам класса.</p>

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	<p>ОПК-3: 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.</p> <p>ОПК-4: 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>ОПК-3: 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>ОПК-4: 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>ОПК-3: 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>

	ОПК-4: 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-3: 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-4: Не понимает принципы инкапсуляции и правила определения доступа к элементам класса. Не умеет реализовывать стандартные интерфейсы и использовать стандартные алгоритмы.

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	ОПК-3: Знать: <ul style="list-style-type: none">● синтаксис языка программирования Java;● принципы объектно-ориентированного программирования. Уметь: <ul style="list-style-type: none">● выполнять отладку и тестирование программы.	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	ОПК-3: Знать: <ul style="list-style-type: none">● синтаксис языка программирования Java;● принципы объектно-ориентированного программирования. Владеть навыками: <ul style="list-style-type: none">● работы с научно- технической литературой и технической документацией Java и JavaFX.	0 баллов – студент не знает определение и принципы работы элементов языка программирования Java; 1 балл – студент в основном описал и проанализировал элементы языка

		<p>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)</p>
3	<p>ОПК-4: Знать: <ul style="list-style-type: none"> ● синтаксис языка программирования Java и библиотеки JavaFX. Уметь: <ul style="list-style-type: none"> ● использовать методы защиты данных. </p>	<p>0 баллов – студент не знает определение и принципы работы элементов языка программирования Java и библиотеки JavaFX; 1 балл – студент в основном описал и проанализировал элементы библиотеки JavaFX с учетом технологий программирования</p>

		и методов защиты данных, возможно с небольшими неточностями или не полностью; 2 балла – студент полностью описал и проанализировал элементы библиотеки JavaFX с учетом технологий программирования и методов защиты данных
4	<p>ОПК-3:</p> <p>Знать:</p> <ul style="list-style-type: none"> ● основы теории алгоритмов и ее применения; ● синтаксис языка программирования Java; <p>Уметь:</p> <ul style="list-style-type: none"> ● разрабатывать программы на языке Java; 	<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</p>

		<p>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>ОПК-3: Знать:</p> <ul style="list-style-type: none"> ● разрабатывать приложения в рамках архитектуры «Модель-Вид-Контроллер». ● принципы объектно-ориентированного программирования. <p>Уметь:</p> <ul style="list-style-type: none"> ● создавать графические пользовательские приложения с использованием JavaFX; 	<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 присутствуют ошибки, в реализации алгоритмов или существенные ошибки в</p>

		применении средств языка программирования 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»
	ОПК-4: Уметь: ● использовать методы защиты данных;	0 баллов – модель решения задачи построена не верно; 1 балл – правильно построена модель решения задачи с учетом защиты данных и организации связей между элементами программы

Максимальное количество баллов по ОПК-3 – 9 баллов

Максимальное количество баллов по ОПК-4 – 3 балла.

Набранное количество баллов соответствует оценке за выполнение экзаменационной работы:

- менее 4 баллов по ОПК-3 или менее 1 балла по ОПК-4 — оценка «неудовлетворительно»,

- не менее 4 баллов по ОПК-3 и не менее 1 балла по ОПК-4, в общей сумме от 5 до 7 баллов — оценка «удовлетворительно», пороговый уровень формирования компетенции,

- не менее 6 баллов по ОПК-3 и не менее 2 баллов по ОПК-4, в общей сумме от 8 до 10 баллов — оценка «хорошо», продвинутый уровень формирования компетенции,

- от 8 до 9 баллов по ОПК-3 и не менее 3 баллов по ОПК-4, в общей сумме от 11 до 12 баллов — оценка «отлично», высокий уровень формирования компетенции,.

Методические указания по выставлению итоговой оценки за экзамен.

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 самостоятельные 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 самостоятельные работы на положительную оценку, возможно не полностью в срок. При выполнении этого условия оценка за экзаменационную работу считается итоговой.

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Общепрофессиональные компетенции						
ОПК-3	Самостоятельная работа Контрольная работа. Экзамен.	1-18	<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; • выполнять отладку и тестирование программы; • разрабатывать приложения в 	<ol style="list-style-type: none"> 1. Знать синтаксис основных конструкций языка Java 2. Знать алгоритмы хранения и обработки данных 3. Знать основные принципы объектно-ориентированного программирования. Уметь разрабатывать классы, описывать подходящие поля и методы 4. Уметь разрабатывать программы на языке Java, реализующие простые алгоритмы 5. Уметь создавать графическое 	<ol style="list-style-type: none"> 1. Знать синтаксис основных конструкций языка Java. Использовать наиболее удобные конструкции для реализации соответствующих алгоритмов. 2. Знать алгоритмы хранения и обработки данных. Знать их преимущества и недостатки. 3. Знать основные принципы объектно-ориентированного программирования. Уметь проектировать приложение из 	<ol style="list-style-type: none"> 1. Знать синтаксис конструкций языка Java. Использовать наиболее удобные конструкции для реализации соответствующих алгоритмов. 2. Знать алгоритмы хранения и обработки данных. Знать их преимущества и недостатки. Использовать только максимально подходящие алгоритмы для поставленной задачи. 3. Знать принципы объектно-

		<p>рамках архитектуры «Модель-Вид-Контроллер».</p> <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. 	<p>приложение из небольшого количества компонентов</p> <p>6. Уметь выполнять отладку программы</p> <p>7. Владеть навыками работы в одной из сред разработки программного обеспечения</p>	<p>нескольких классов</p> <p>4. Уметь разрабатывать программы на языке Java, реализующие алгоритмы различной сложности</p> <p>5. Уметь создавать графическое приложение. Разрабатывать структуру классов на основе архитектурного шаблона Модель-Вид-Контроллер</p> <p>6. Уметь выполнять отладку программы. Оптимизировать работу алгоритмов</p> <p>7. Владеть навыками работы в одной из сред разработки программного обеспечения. Уверенно пользоваться различными инструментами разработки программных приложений</p> <p>8. Работать с документацией.</p>	<p>ориентированного программирования. Уметь проектировать приложение из нескольких классов, разрабатывать иерархию наследования</p> <p>4. Уметь разрабатывать программы на языке Java, реализующие алгоритмы различной сложности. Оптимизировать работу существующих алгоритмов</p> <p>5. Уметь создавать сложное многооконное графическое приложение. Разрабатывать структуру классов на основе архитектурного шаблона Модель-Вид-Контроллер</p> <p>6. Уметь выполнять отладку программы. Пользоваться инструментами для автоматического тестирования программы</p> <p>7. Владеть навыками работы в нескольких средах разработки</p>
--	--	--	--	---	---

						программного обеспечения. Уверенно пользоваться различными инструментами разработки программных приложений 8. Работать с документацией.
ОПК -4	Самостоятельная работа Контрольная работа. Экзамен.	1-18	<p>Знать:</p> <ul style="list-style-type: none"> • синтаксис языка программирования Java и библиотеки JavaFX; • принципы объектно-ориентированного программирования. <p>Уметь:</p> <ul style="list-style-type: none"> • разрабатывать программы на языке Java; • использовать методы защиты данных; • выполнять отладку программы. <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. 	<ol style="list-style-type: none"> 1. Знать синтаксис основных конструкций языка Java 2. Знать алгоритмы хранения и обработки данных 3. Знать основные принципы объектно-ориентированного программирования. <p>Уметь разрабатывать классы, описывать подходящие поля и методы</p> <ol style="list-style-type: none"> 4. Уметь разрабатывать программы на языке Java, реализующие простые алгоритмы 5. Уметь создавать графическое приложение из небольшого количества компонентов 6. Уметь выполнять 	<ol style="list-style-type: none"> 1. Знать синтаксис основных конструкций языка Java. <p>Использовать наиболее удобные конструкции для реализации соответствующих алгоритмов.</p> <ol style="list-style-type: none"> 2. Знать алгоритмы хранения и обработки данных. Знать их преимущества и недостатки. 3. Знать основные принципы объектно-ориентированного программирования. <p>Уметь проектировать приложение из нескольких классов</p> <ol style="list-style-type: none"> 4. Уметь разрабатывать программы на языке Java, реализующие 	<ol style="list-style-type: none"> 1. Знать синтаксис конструкций языка Java. Использовать наиболее удобные конструкции для реализации соответствующих алгоритмов. 2. Знать алгоритмы хранения и обработки данных. Знать их преимущества и недостатки. Использовать только максимально подходящие алгоритмы для поставленной задачи. 3. Знать принципы объектно-ориентированного программирования. Уметь проектировать приложение из

			<p>отладку программы</p> <p>7. Владеть навыками работы в одной из сред разработки программного обеспечения</p>	<p>алгоритмы различной сложности</p> <p>5. Уметь создавать графическое приложение. Разрабатывать структуру классов на основе архитектурного шаблона Модель-Вид-Контроллер</p> <p>6. Уметь выполнять отладку программы. Оптимизировать работу алгоритмов</p> <p>7. Владеть навыками работы в одной из сред разработки программного обеспечения. Уверенно пользоваться различными инструментами разработки программных приложений</p> <p>8. Работать с документацией.</p>	<p>нескольких классов, разрабатывать иерархию наследования</p> <p>4. Уметь разрабатывать программы на языке Java, реализующие алгоритмы различной сложности. Оптимизировать работу существующих алгоритмов</p> <p>5. Уметь создавать сложное многооконное графическое приложение. Разрабатывать структуру классов на основе архитектурного шаблона Модель-Вид-Контроллер</p> <p>6. Уметь выполнять отладку программы. Пользоваться инструментами для автоматического тестирования программы</p> <p>7. Владеть навыками работы в нескольких средах разработки программного обеспечения. Уверенно пользоваться различными</p>
--	--	--	--	---	---

						инструментами разработки программных приложений 8. Работать с документацией.
--	--	--	--	--	--	---

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объеме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;

- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;

- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Показатели и критерии, используемые при выставлении оценки подробно описаны в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций».

Высокий уровень формирования компетенций соответствует оценке «отлично» за самостоятельные, контрольные работы и экзаменационную работу.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за самостоятельные, контрольные работы и экзаменационную работу.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за самостоятельные, контрольные работы и экзаменационную работу.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Приложение №2 к рабочей программе дисциплины «Языки и методы программирования»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Языки и методы программирования» 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 самостоятельные работы различного объема. Примеры решения задач разбираются на лекциях, кроме того проводятся лабораторные занятия на сопутствующих практикумах ЭВМ: «Практикум на ЭВМ по объектно-ориентированному программированию», «Практикум на ЭВМ по программированию». 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. Кроме того, материал курса соответствует лабораторным работам практикумов: «Практикум на ЭВМ по объектно-ориентированному программированию», «Практикум на ЭВМ по программированию».

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 «Языки и методы программирования» 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.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее

востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека [«Книгообеспеченность»](#) (http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека [«Книгообеспеченность»](#) доступна в сети университета и через Личный кабинет.