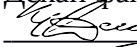


**МИНОБРНАУКИ РОССИИ**  
**Ярославский государственный университет им. П.Г. Демидова**

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**  
02.03.02 Фундаментальная информатика и информационные технологии

**Профиль**  
«Информатика и компьютерные науки»

**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 (на примере JavaFX), 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 объектно-ориентированного программирования», проявлять настойчивость, целеустремленность и инициативу в процессе обучения.

Знания и навыки, полученные при изучении практикума, используются учащимися при изучении последующих дисциплин базовой и вариативной частей ОП, а именно «Программная инженерия», «Современные мобильные платформы и сервисы», а также для продолжения обучения в магистратуре по направлению Фундаментальная информатика и информационные технологии.

### 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
<b>Общепрофессиональные компетенции</b>		
ОПК-4 Способен участвовать в разработке технической документации программных продуктов и комплексов с использованием стандартов, норм и правил, а также в управлении проектами создания информационных систем на стадиях жизненного цикла	ОПК-4.1 Знает основные принципы организации и разработки программ сложной структуры ОПК-4.2 Знает стандарты разработки программных проектов и технической документации	Знать: • основы теории алгоритмов и ее применения; • синтаксис языка программирования Java; • принципы объектно-ориентированного программирования. Уметь: • разрабатывать программы на языке Java; • создавать графические пользовательские приложения с использованием JavaFX; • выполнять отладку программы; • разрабатывать приложения в рамках архитектуры «Модель-Вид-Контроллер».

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

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

Общая трудоемкость дисциплины составляет 4 зач. ед., 144 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Се мес тр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)					Формы текущего контроля успеваемости	
			лек ции	пра кти чес кие	лаб ора тор ные	кон сул ьта ции	атт ест аци онн ые исп ыта ния	само стоя тельн ая рабо та	Форма промежуточной аттестации (по семестрам)
			Контактная работа						
1.	Контейнеры, коллекции	4	4		4			4	
2.	Графические пользовательские приложения	4	4		4			2	
3.	Введение в JavaFX	4	4		4			3	
4.	Архитектура создания пользовательских приложений: Модель Вид Контроллер.	4	4		4	1		4	Контрольная работа
5.	Основные элементы сцены	4	4		4	1		4	Задания для самостоятельной работы
6.	Взаимодействие между элементами приложения. События	4	4		4	1		4	
7.	Организация ввода данных. Диалоговые окна.	4	4		4	1		4	
8.	Главное окно приложения.	4	4		4	1		4	
9.	Разработка декларативного	4	2		2	1		5	Задания для самостоятельной работы

	интерфейса с помощью FXML							
					2			Экзамен
	<b>Всего за 4 семестр</b>	<b>34</b>		<b>34</b>	<b>8</b>		<b>34</b>	<b>Экзамен</b>
	<b>Всего</b>	<b>34</b>		<b>34</b>	<b>8</b>		<b>34</b>	

### Содержание разделов дисциплины:

1. Контейнеры, коллекции. Основные классы и интерфейсы. Итераторы. Последовательности и множества. Свойства и преимущества контейнеров.
2. Графические пользовательские приложения. Инструменты разработки: фреймворк, среда разработки, система контроля версий.
3. Введение в JavaFX. Понятие графической библиотеки. Документация.
4. Архитектура создания пользовательских приложений: Модель Вид Контроллер. Основные элементы архитектурного шаблона. Связи между элементами.
5. Основные элементы сцены. Элементы отображения, элементы управления, невизуальные элементы. Свойства элементов. Компановка элементов.
6. Взаимодействие между элементами приложения. События. Обработчики событий. Лямбда-выражения.
7. Организация ввода данных. Диалоговые окна. Стандартные диалоги. Создание пользовательских диалоговых окон.
8. Главное окно приложения. Меню, виды меню. Расположение элементов. Обработка событий.
9. 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-НЕХТ" (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>

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](http://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

#### **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 и практических занятий (семинаров);

- учебные аудитории для проведения групповых и индивидуальных консультаций,

- учебные аудитории для проведения текущего контроля и промежуточной аттестации;  
-помещения для самостоятельной работы;  
-помещения для хранения и профилактического обслуживания технических средств обучения.

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

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 лабораторных занятий и 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 «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**

<b>Оценка</b>	<b>Критерии</b>
Отлично	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. Умеет создавать однооконное графическое приложение в среде разработки. Поясняет код и изменяет его при необходимости. Использует техническую документацию при разработке программы, материалы лекций и другие источники информации.
Хорошо	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	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	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 «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	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. 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.
Хорошо	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. 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	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. 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	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. 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.

### **Типовой вариант контрольной работы**

#### 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	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	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	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	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 зачету

1. Контейнеры.
  - Виды контейнеров.
  - Свойства контейнеров.
  - Хранение объектов библиотечных классов и пользовательских.
  - Итераторы.
  - Алгоритмы.
2. Основные элементы графических пользовательских приложений JavaFX. Сцена  
Элементы отображения информации  
Управляющие элементы  
Обработка события нажатия кнопки  
Расположение элементов на сцене  
Однооконное приложение
3. Архитектура графического пользовательского приложения Модель-вид-контроллер
  - Разработка модели.
  - Активная и пассивная модель
  - Отображение данных.
  - Взаимодействие классов «модель» и «вид»
  - Функции контроллера
4. Диалоги.
  - Разработка диалога для ввода данных.
  - Проверка вводимых данных
  - Стандартные диалоги
5. Меню, контекстное меню
6. Обработка событий
  - События мыши
  - События клавиатуры

### Методические указания по выставлению зачета

Зачет выставляется по результатам выполнения всех самостоятельных и контрольных работ на оценку не ниже удовлетворительно. Работы выполняются и сдаются в течение семестра последовательно в процессе освоения материала или в исключительных случаях на зачете. В случае необходимости преподаватель в ходе сдачи работ может провести беседу по вопросам к зачету, связанным в первую очередь с тематикой работы.

## **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.

### **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 объёме 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 «зачтено», «незачтено».

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 и контрольные работы.

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

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

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.

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

Оценка «незачтено» 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 лабораторные работы различного объема, а так же небольшие задачи связанные с исправлением ошибок и доработкой программ. Примеры решения задач разбираются на лекциях, кроме того проводятся лабораторные занятия 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. Также проводятся консультации (при необходимости) по разбору заданий, которые вызвали затруднения.

В конце изучения дисциплины студенты сдают зачет. Зачет выставляется по итогам работы, которая включает в себя выполнение самостоятельных и контрольных работ.

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](http://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](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](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](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.