

Минобрнауки России
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ
Заведующий кафедрой
ПиИТ



*проф. Махортов
С.Д.*

подпись, расшифровка подписи

05.03.2024 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.О.20 Объектно-ориентированное программирование

1. Код и наименование направления подготовки/специальности:

09.03.02 Информационные системы и технологии

2. Профиль подготовки/специализация:

Инженерия информационных систем и технологий

3. Квалификация (степень) выпускника: Бакалавриат

4. Форма обучения: Очная

5. Кафедра, отвечающая за реализацию дисциплины: Кафедра программирования и информационных технологий

6. Составители программы: Самойлов Николай Константинович, ст. преп.

7. Рекомендована: НМС ФКН, протокол № 5 от 05.03.2024

8. Учебный год: 2025-2026

Семестр: 3

9. Цели и задачи учебной дисциплины:

Цели учебной дисциплины:

- изучение объектно ориентированного программирования на платформе Java.
- изучение базовых возможностей работы на платформе Java.

Задачи учебной дисциплины:

- овладение базовыми возможностями платформы Java;
- знакомство с принципами объектно-ориентированного программирования; освоение инструментария коллекций, ввода-вывода, Stream API платформы Java.

10. Место учебной дисциплины в структуре ООП:

Дисциплина относится к обязательной части блока Б1. Для ее успешного освоения необходимы знания из курсов "Введение в программирование", "Дискретная математика". Данная дисциплина является предшествующей для дисциплины "Язык программирования Java".

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников):

Код и название компетенции	Код и название индикатора компетенции	Знания, умения, навыки
ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;	ОПК-1.1 Знает основы математики, физики, вычислительной техники и программирования	Знать: теорию графов, теорию множеств, основы дискретной математики. Уметь: применять теоретические знания при реализации программ. Владеть: средствами разработки и отладки для платформы Java.
	ОПК-2.1 Знает современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	Знать: основные возможности платформы Java 8, средства автоматизации сборки проектов Apache Maven. Уметь: применять знания платформы Java 8 и Apache Maven при реализации программ. Владеть: средствами разработки и отладки для платформы Java используя средства автоматизации сборки проектов.
ОПК-2 Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;	ОПК-2.2 Умеет выбирать современные информационные технологии и программные средства, в том числе	Знать: статус различных версий платформы Java и их различия, средства

	<p>отечественного производства, при решении задач профессиональной деятельности</p>	<p>автоматизации сборки проектов Apache Maven.</p> <p>Уметь: устанавливать и конфигурировать программные комплексы платформы Java и Apache Maven.</p> <p>Владеть: установки и конфигурации разработки для платформы Java.</p>
	<p>ОПК-2.3 Имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности</p>	<p>Знать: основные возможности платформы Java 8, средства автоматизации сборки проектов Apache Maven.</p> <p>Уметь: применять знания платформы Java 8 и Apache Maven при реализации программ.</p> <p>Владеть: средствами разработки и отладки для платформы Java используя средства автоматизации сборки проектов.</p>
<p>ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;</p>	<p>ОПК-6.1 Знает методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий</p>	<p>Знать: принципы объектноориентированного программирования, базовые возможности платформы Java.</p> <p>Уметь: выстраивать иерархию наследования.</p> <p>Владеть: средствами разработки и отладки для платформы Java.</p>
	<p>ОПК-6.2 Умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий</p>	<p>Знать: общие концепции алгоритмов, базовые возможности платформы Java.</p> <p>Уметь: формулировать бизнеслогику исходя из поставленной задачи.</p> <p>Владеть: методами алгоритмизации на платформе Java.</p>
	<p>ОПК-6.3 Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>	<p>Знать: общие концепции алгоритмов, отладки программ.</p> <p>Уметь: реализовывать программы на платформе Java.</p>

Владеть: средствами разработки и отладки на платформе Java.

12. Объем дисциплины в зачетных единицах/час: 3/108

Форма промежуточной аттестации: экзамен

13. Виды учебной работы

Вид учебной работы	Семестр 3	Всего
Аудиторные занятия	48	48
Лекционные занятия	34	34
Практические занятия		
Лабораторные занятия	16	16
Самостоятельная работа	22	22
Промежуточная аттестация	36	36
Всего	108	108

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1. Лекции			
1.1	Основы объектноориентированного программирования	Принципы объектно ориентированного программирования, абстракция сущностей.	Онлайн курс "Объектноориентированное программирование"
1.2	Базовые возможности языка Java	Классы, объекты, модификаторы, модификаторы доступа, сравнение объектов.	Онлайн курс "Объектноориентированное программирование"
1.3	Перечисления в Java	Перечисления в языке Java, добавление состояния и методов в перечисление.	Онлайн курс "Объектноориентированное программирование"
1.4	Исключения в Java	Понятие исключения, исключения в Java, перехват исключений, виды исключений.	Онлайн курс "Объектноориентированное программирование"
1.5	Обобщенные типы в Java	Обобщенные типы в Java, стирание типов, принцип подстановки, символы подстановки, Get/Putпринципы, wildcard capture, ковариантность обобщенных типов.	Онлайн курс "Объектноориентированное программирование"
1.6	Коллекции в Java	Списки, множества, соответствия, стек, очередь. Обход коллекции, удаление элементов коллекции,	Онлайн курс "Объектноориентированное программирование"

		сравнение коллекций, поиск элемента в коллекции. Особые коллекции: только для чтения, синхронизированные, пустые, вырожденные.	
1.7	Элементы функционального программирования в Java	Понятие лямбда-выражения, понятие функционального интерфейса, сравнение особенностей лямбдавыражения и анонимного класса. Стандартные функциональные интерфейсы, область видимости переменной, ссылка на метод.	Онлайн курс "Объектноориентированное программирование"
1.8	Java Stream API	Внешнее и внутреннее итерирование по коллекции. Промежуточные и терминальные операции, характеристики потоков данных. Создание потоков данных, необязательное значение. Операции peek, map, flatMap, filter. Операция reduce, группировка данных, формирование коллекции из потока данных.	Онлайн курс "Объектноориентированное программирование"
1.9	JSON-сериализация объектов	Понятие сериализации, формат JSON, инструментарий Google Gson, инструментарий FasterXml Jackson.	Онлайн курс "Объектноориентированное программирование"
2. Практические занятия			
2.1	Инструментарий IntelliJ IDEA	Знакомство с инструментарием, создание тестового приложения, освоение возможностей	Онлайн курс "Объектноориентированное программирование"
2.2	Иерархия классов приложения	Реализация классов приложения, скрытие членов класса, типичные ошибки проектирования иерархии классов	Онлайн курс "Объектноориентированное программирование"
2.3	Слоевая архитектура приложения	Разделение приложения на слои: модель, сервисы, контроллеры. Типичные ошибки слоевого проектирования. Реализация бизнес-логики приложения	Онлайн курс "Объектноориентированное программирование"
2.4	Фреймворк Apache Maven	Миграция приложения на Apache Maven. Освоение базового функционала инструментария сборки и поставки	Онлайн курс "Объектноориентированное программирование"

2.5	Реализация библиотеки классов	Реализация библиотеки классов коллекций. Интеграция реализованной библиотеки в тестовый проект	Онлайн курс "Объектноориентированное программирование"
3. Лабораторные работы			
3.1	Реализация структуры классов приложения	Реализация статической части приложения - структуры классов, реализация базовых элементов бизнес-логики, разделение классов приложения на слои.	
3.2	Реализация бизнес логики приложения.	Реализация бизнес-логики приложения: использование лямбда-выражений, Stream API. Реализация сериализации/десериализации в JSON состояния приложения в любой момент времени жизни приложения.	
3.3	Реализация библиотеки классов	Реализация библиотеки классов, имплементирующую функционал коллекций, иерархий или графов.	

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
1	Основы объектноориентированного программирования	4		2	1	7
2	Базовые возможности языка Java	4		2	1	7
3	Перечисления в Java	3		1	2	6
4	Исключения в Java	3		1	2	6
5	Обобщенные типы в Java	4		2	4	10
6	Коллекции в Java	4		2	4	10
7	Элементы функционального программирования в Java	4		2	2	8
8	Java Stream API	4		2	4	10
9	JSON-сериализация объектов	4		2	2	8
		34		16	22	72

14. Методические указания для обучающихся по освоению дисциплины

Работа с конспектами лекций и презентационным материалом; выполнение практических заданий; выполнение лабораторных заданий; подготовка к заданиям текущей аттестации.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

№ п/п	Источник
1	Эккель, Брюс. Философия Java = Thinking in Java / Брюс Эккель ; [пер. с англ. Е. Матвеева] . — 4-е полное изд. — Москва : Вильямс, 2017 .— 1165 с.
2	Sharan, K. Beginning Java 8 Language Features, Apress, 2014, 690 p.
3	Warburton, R, Java 8 Lambdas, OReilly, 2014, 182 p.

б) дополнительная литература:

№ п/п	Источник
1	Bloch, J. Effective Java, Third Edition, Addison-Wesley, 2018, 413 p.
2	Subramanian, V. Functional Programming in Java, Pragmatic Bookshelf, 2014, 185 p.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
1	www.lib.vsu.ru – ЗНБ ВГУ
2	Java Platform Standard Edition 8 Documentation https://docs.oracle.com/javase/8/docs/
3	Java8 Tutorial https://www.tutorialspoint.com/java8/index.htm

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	Java8 Tutorial https://www.tutorialspoint.com/java8/index.htm

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости):

IDE IntelliJ IDEA 2020

18. Материально-техническое обеспечение дисциплины:

Мультимедийная лекционная аудитория (корп.1б, ауд. № 505):

ПК-Intel-i3, рабочее место преподавателя: проектор, видеоконмутатор, микрофон, аудиосистема, специализированная мебель, выход в Интернет, доступ к фондам учебно-методической документации и электронным изданиям.

Компьютерный класс (один из №1-4 корп. 1а, ауд. № 382-385):

ПК-Intel-i3 16 шт., специализированная мебель: доска маркерная 1 шт., столы 16 шт., стулья 33 шт.; доступ к фондам учебно-методической документации и электронным изданиям, доступ к электронным библиотечным системам, выход в Интернет.

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
1	Основы объектноориентированного программирования	ОПК-1	ОПК-1.1	Практические задания размещены на портале "Электронный университет ВГУ" https://edu.vsu.ru/user/index.php?id=4115
2	Базовые возможности языка Java	ОПК-6	ОПК-6.1, ОПК-6.2, ОПК-6.3	Практические задания размещены на портале "Электронный университет ВГУ" https://edu.vsu.ru/user/index.php?id=4115
3	Обобщенные типы в Java	ОПК-1, ОПК-6	ОПК-1.1, ОПК-6.1, ОПК-6.2, ОПК-6.3	Практические задания размещены на портале "Электронный университет ВГУ" https://edu.vsu.ru/user/index.php?id=4115
№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
4	Коллекции в Java	ОПК-1, ОПК-6	ОПК-1.1, ОПК-6.1, ОПК-6.2, ОПК-6.3	Практические задания размещены на портале "Электронный университет ВГУ" https://edu.vsu.ru/user/index.php?id=4115
5	Элементы функционального программирования в Java	ОПК-1, ОПК-6	ОПК-1.1, ОПК-6.1, ОПК-6.2, ОПК-6.3	Практические задания размещены на портале "Электронный университет ВГУ" https://edu.vsu.ru/user/index.php?id=4115
6	Java Stream API	ОПК-2, ОПК-6	ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-6.1, ОПК-6.2	Практические задания размещены на портале "Электронный университет ВГУ" https://edu.vsu.ru/user/index.php?id=4115
7	JSON-сериализация объектов	ОПК-2, ОПК-6	ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-6.1, ОПК-6.2	Практические задания размещены на портале "Электронный университет ВГУ" https://edu.vsu.ru/user/index.php?id=4115

Промежуточная аттестация

Форма контроля - экзамен, Контрольная работа

Оценочные средства для промежуточной аттестации

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств: рубежные аттестации лабораторные работы. Перечень заданий для лабораторных работ соответствует темам занятий. Решение каждого задания должно быть доведено до компьютерной реализации.

20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств: собеседование по билетам к экзамену.

Перечень вопросов к экзамену:

1. Инкапсуляция.
2. Управление доступом, модификаторы доступа.
3. Понятие класса.
4. Конструкторы и блоки инициализации.
5. Наследование.
6. Upcasting, Downcasting, оператор instanceof.
7. Модификатор static.
8. Модификатор final.
9. Корень иерархии типов - класс Object.
10. Полиморфизм.
11. Абстрактный класс, абстрактный метод.
12. Понятие интерфейса, члены интерфейса.
13. Реализация интерфейса.
14. Сравнение объектов на равенство, методы equals(), hashCode().
15. Класс Objects.
16. Перечисления.
17. Понятие исключения. Перехват и обработка исключения.
18. Иерархия типов исключений.
19. Обработка нескольких типов исключений.
20. Проверяемые и непроверяемые исключения.
21. Выброс исключения.
22. Объявление собственных классов исключений.
23. Понятие обобщенного типа.

24. Стирание типов.
25. Принцип подстановки.
26. Get-принцип.
27. Put-принцип.
28. Захват символа подстановки.
29. Типы коллекций, итератор.
30. Списки.
31. Множества.
32. Очередь и дек.
33. Очередь с приоритетами.
34. Соответствие/словарь.
35. Обход коллекции, удаление элементов коллекции.
36. Сравнение коллекций.
37. Компаратор. Сортировка коллекции.
38. Поиск элемента в коллекции.
39. Специализированные коллекции: только для чтения, синхронизированные, пустые, вырожденные.
40. Понятие лямбда-выражения.
41. Функциональный интерфейс.
42. Сравнение особенностей лямбда-выражений и анонимных классов.
43. Анонимная реализация интерфейса.
44. Стандартные функциональные интерфейсы.
45. Захват переменной лямбда-выражением.
46. Лямбда-синтаксис для компараторов.
47. Ссылки на методы.
48. Внешнее/внутреннее итерирование.
49. Понятие потока данных (Stream API). Характеристики потока данных.
50. Промежуточные и терминальные операции над потоком данных.
51. Способы создания потока данных.
52. Необязательное значение.
53. Операции peek, forEach.
54. Операции map, flatMap.
55. Операция filter.
56. Формирование коллекции из потока данных.
57. Операции reduce, groupingBy.
58. Понятие сериализации, десериализации.

59. JSON-формат.

60. Инструментарий Google Gson.

61. Инструментарий FasterXml Jackson.

20.3 Приведённые ниже задания рекомендуется использовать при проведении диагностических работ для оценки остаточных знаний по дисциплине

1) тестовые задания – 1 балл

- Конструктор создаёт экземпляр класса?
 - да
 - нет
- Правда ли что в языке Java поддерживается множественное наследование?
 - да
 - нет
- Какие требования предъявляются к абстрактному классу?
 - Объявление класса должно содержать ключевое слово **abstract**
 - Абстрактный класс должен иметь хотя бы один абстрактный метод
 - Абстрактный класс должен содержать несколько абстрактных методов
- Перечислите различия между интерфейсом и абстрактным классом
 - Абстрактный класс может содержать поля, а интерфейс нет
 - Абстрактный класс может быть наследником только одного класса, а интерфейс может быть наследником нескольких интерфейсов
 - Абстрактный класс может содержать любые поля, а интерфейс только **public static final** поля
 - Абстрактный класс может содержать статические методы, а интерфейс нет
 - Абстрактный класс может иметь конструкторы, а интерфейс нет
- Для сравнения объектов на равенство нужно использовать?
 - Оператор ==
 - Метод equals
- Модификатор **static** используется для объявления константных полей?
 - да
 - нет
- Непроверяемые исключения являются непосредственными наследниками класса:
 - Exception
 - RuntimeException
 - Error
- Членами интерфейса могут быть:
 - Абстрактные методы
 - Экземплярные методы
 - Статические методы
 - Статические инициализаторы
 - Экземплярные поля

Ответы на вопросы

Номер вопроса	Ответ (буква)
1.	b
2.	b
3.	a
4.	b, c, e
5.	b
6.	b
7.	b
8.	a, b, c

2) задания с коротким ответом – 2 балла

- Сколько существует принципов объектно-ориентированного программирования
Ответ (число)

2. Какие типы из пакета java.util реализуют функционал динамического списка

Ответ (строка)

3. Какой утилитный класс позволяет создавать экземпляры пустых коллекций, коллекций только для чтения?

Ответ (строка)

4. Какие классы из пакета java.util реализуют функционал словаря?

Ответ (строка)

5. Какую аннотацию необходимо использовать при перекрытии виртуальных методов?

Ответ (строка)

Ответы на вопросы

Номер вопроса	Ответ
1	3
2	LinkedList, ArrayList
3	Collections
4	HashMap, LinkedHashMap, TreeMap
5	Override

3) задания с развернутым ответом

1. Необходимо реализовать классы Student и Group. Класс Student содержит поля: имя, возраст, пол. Класс Group содержит поля: наименование, список студентов группы, специализация. (Для реализации классов можно использовать инструментарий Lombok)

Вопросы к заданию:

1. Какие типы можно использовать для объявления поля «возраст»?
2. Какой тип необходимо использовать для объявления поля «пол»?
3. Какой тип можно использовать для объявления поля «список студентов группы»?

2. Класс Student содержит поля: номер СНИЛС, фамилия, имя, возраст, пол. Необходимо реализовать механизм сравнения студентов на равенство отличный от стандартного.

Вопросы к заданию:

1. По какому полю/полям можно уникально идентифицировать экземпляры класса Student?
2. Какие методы необходимо перекрыть для реализации механизма сравнения?
3. Какие требования предъявляются к методам, которые используются для сравнения объектов на равенство?

3. Необходимо реализовать функционал динамического списка

Вопросы к заданию:

1. Какой интерфейс должен реализовывать класс?
2. Назовите минимум 5 методов, которые должен реализовывать класс?
3. Каким образом можно реализовать хранение данных динамического списка?

4. Необходимо реализовать структуру типов, которая описывает отношения между сотрудниками ИТ-компании. Каждый сотрудник имеет имя, возраст и зарплату. В организации существуют следующие должности: разработчик (пишет программы), тестировщик (тестирует программы), аналитик (создает задачи). Любой сотрудник помимо основной должности может иметь дополнительные должностные обязанности такие как: менеджер сборок, скрам-мастер, владелец продукта. (Для реализации классов можно использовать инструментарий Lombok)

Вопросы к заданию:

1. Какие структуры необходимо использовать для реализации?
2. Как можно реализовать тип разработчик, который также является менеджером сборок?
3. Как можно реализовать тип аналитик, который также является скрам-мастером и владельцем продукта?

Ответы на вопросы

Номер вопроса	Ответ
1.	<p>На первый вопрос: byte, Byte, short, Short, int Integer</p> <p>На второй вопрос: enum</p> <p>На третий вопрос: List</p>
2.	<p>На первый вопрос По полю СНИЛС</p> <p>На второй вопрос equals(), hashCode()</p> <p>На третий вопрос Если a.equals(b) то b.equals(a) Если a.equals(b) то a.hashCode() == b.hashCode() Если a.hashCode() != b.hashCode(), то a и b не равны в по equals</p>
3.	<p>На первый вопрос Интерфейс List</p> <p>На второй вопрос Методы add(), remove(), contains(), size(), isEmpty(), get()</p> <p>На третий вопрос Используя массив или связный список</p>
4.	<p>На первый вопрос Классы и интерфейсы</p> <p>На второй вопрос <pre>public class Developer extends Employee implements BuildManager { public String writeProgram() {} @Override public void buildProgram() {} }</pre> </p> <p>На третий вопрос <pre>public class Analyst extends Employee implements ScramMaster, ProductOwner { public Task createTask() {} @Override public void holdMeeting() {} @Override public void manageProject() {} }</pre> </p>

Критерии оценивания	Шкала оценок (в баллах)
Обучающийся отвечает на все вопросы правильно.	3 балла
Обучающийся отвечает на все вопросы.	2 балла

Допускаются незначительные неточности.	
Обучающийся отвечает не на все вопросы. Ответ не содержит грубых ошибок.	1 балл
Обучающийся отвечает не на все вопросы. Присутствуют грубые ошибки или неточности.	0 баллов