

Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет имени В.А. Бондаренко»

Кафедра математики и цифровых технологий

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«Б1.Д.В.2 Объектно-ориентированные языки и системы»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.02 Информационные системы и технологии
(код и наименование направления подготовки)

Анализ данных и машинное обучение

(наименование направленности (профиля) образовательной программы)

Квалификация

Бакалавр

Форма обучения

Очная

Год набора 2026

Рабочая программа дисциплины «Б1.Д.В.2 Объектно-ориентированные языки и системы» рассмотрена и утверждена на заседании кафедры

Кафедра математики и цифровых технологий

наименование кафедры

протокол № 7 от " 19 " 03 2026 г.

Заведующий кафедрой

Кафедра математики и цифровых технологий

наименование кафедры

подпись

А.Е. Шухман

расшифровка подписи

Исполнители:

Старший преподаватель кафедры МЦГ

должность

подпись

И.В. Минина

расшифровка подписи

должность

подпись

расшифровка подписи

СОГЛАСОВАНО:

Председатель методической комиссии по направлению подготовки

09.03.02 Информационные системы и технологии

код наименование

личная подпись

А.Е. Шухман

расшифровка подписи

Заведующий отделом формирования фонда и научной обработки документов

личная подпись

С.А. Биктимирова

расшифровка подписи

Уполномоченный по качеству института

личная подпись

С.Н. Морозова

расшифровка подписи

№ регистрации _____

© Минина И.В., 2026
© ОГУ, 2026

1 Цели и задачи освоения дисциплины

Цель (цели) освоения дисциплины: формирование профессиональных компетенций в области проектирования и разработки программного обеспечения с использованием объектно-ориентированных языков и систем.

Задачи:

- формирование представлений об общей методологии и средствах технологии объектно-ориентированного программирования;
- освоение технологии объектно-ориентированного программирования;
- получить представление об истории создания, появления и развития платформ языков программирования Python и Java;
- изучить основные свойства, средства и утилиты платформ Python и Java, возможности данных языков;
- научиться разрабатывать иерархию классов на языках Python и Java для заданной предметной области;
- научиться применять современные системы и технологии разработки с использованием возможностей отладки, обработки исключений и использования шаблонов.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательным дисциплинам (модулям) вариативной части блока Д «Дисциплины (модули)»

Пререквизиты дисциплины: *Б1.Д.Б.12 Физика, Б1.Д.Б.14 Информатика, Б1.Д.Б.17 Математическая логика и дискретная математика*

Постреквизиты дисциплины: *Б1.Д.В.5 Основы теории принятия решений, Б1.Д.В.11 Распределенные информационные системы*

3 Требования к результатам обучения по дисциплине

Процесс изучения дисциплины направлен на формирование следующих результатов обучения

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
ПК*-5 Способен разрабатывать требования, проектировать программное обеспечение информационных систем, выполнять интеграцию и проверку работоспособности программных модулей и компонент	ПК*-5-В-1 Умеет разрабатывать средства (методические, информационные, математические, алгоритмические и программные) для реализации информационных технологий в задачах принятия решений в информационных системах среднего и крупного масштаба и сложности	<u>Знать:</u> – основные понятия, принципы и модели ООП - пользовательские типы данных – классы, объект, объявление и реализацию класса, данные, методы; доступ, наследование свойств, виртуальные функции; – алгоритмы и методы проектирования объектно-ориентированных программ; – средства и подходы конструирования собственных пользовательских классов; - объектно-ориентированные языки, язык Python и Java; – правила организации, методы и средства программирования интерфейса; системы программирования; объектно-ориентированные системы; <u>Уметь:</u> – применять при разработке, анализе и

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
		проверке программ принципы объектно-ориентированного программирования; – определять классы, создавать объекты, разрабатывать и использовать конструкторы класса; создавать программы, используя библиотеки классов; – выбрать и использовать алгоритмы и методы проектирования объектно-ориентированных программ; – использовать правила организации и программирования интерфейса; <u>Владеть:</u> – навыками объектно-ориентированного программирования на языках Python и Java; – основными методами и приемами, и средствами разработки объектно-ориентированных программ.

4 Структура и содержание дисциплины

4.1 Структура дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц (216 академических часов).

Вид работы	Трудоемкость, академических часов		
	5 семестр	6 семестр	всего
Общая трудоёмкость	72	144	216
Контактная работа:	53,5	34,25	87,75
Лекции (Л)	18	18	36
Лабораторные работы (ЛР)	34	16	50
Индивидуальная работа и инновационные формы учебных занятий	1		1
Промежуточная аттестация (зачет, экзамен)	0,5	0,25	0,75
Самостоятельная работа: - выполнение курсовой работы (КР); - самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий) - подготовка к лабораторным занятиям; - подготовка к рубежному контролю	18,5 +	109,75	128,25
Вид итогового контроля (зачет, экзамен, дифференцированный зачет)	зачет	зачет	

Разделы дисциплины, изучаемые в 5 семестре

№ раздела	Наименование разделов	Количество часов				
		всего	аудиторная работа			внеауд. работа
			Л	ПЗ	ЛР	
1	Пользовательские функции и основы функционального программирования в Python	22	6		10	6
2	Объектно-ориентированное программирование в Python	26	6		14	6
3	Программирование с графическим интерфейсом в Python	24	6		10	8
	Итого:	72	18		34	20

Разделы дисциплины, изучаемые в 6 семестре

№ раздела	Наименование разделов	Количество часов				
		всего	аудиторная работа			внеауд. работа
			Л	ПЗ	ЛР	
4	Основы процедурного программирования на Java	12	1		1	10
5	Знакомство с компилятором Java и документацией	12	1		1	10
6	Коллекции	13	2		1	10
7	Потоки данных	13	2		1	10
8	Основы объектно-ориентированного программирования	13	1		2	10
9	Наследование и абстракции	13	1		2	10
10	Обработка ошибок и логирование	14	2		2	10
11	Дженерики. Функциональные интерфейсы	14	2		2	10
12	Стримы	14	2		2	10
13	Рефлексия и аннотации	13	2		1	10
14	Unit-тестирование. Java-doc	13	2		1	10
	Итого:	144	18		16	110
	Всего:	216	36		50	130

4.2 Содержание разделов дисциплины

1 Пользовательские функции и основы функционального программирования в Python

Определение пользовательских функций. Передача параметров и возврат результатов. Значения аргументов функции по умолчанию. Произвольный набор аргументов. Именованные аргументы. Правила видимости. Функции, как объекты и замыкания. Декораторы. Итераторы. Генераторы и сопрограммы. Генераторы списков. Выражения-генераторы. Основы декларативного программирования. Оператор Lambda. Атрибуты функций. Выполнение неопределенных функций, командами: eval, exec, comp

2 Объектно-ориентированное программирование в Python

Определение объектно-ориентированной парадигмы. Основные понятия: классы, объекты, наследование. Преимущества и недостатки объектно-ориентированного программирования. Создание классов и объектов в Python. Методы и атрибуты классов. Конструкторы и деструкторы. Принцип наследования и его применение. Переопределение методов и атрибутов. Интерфейсы и абстрактные классы. Полиморфизм и его роль. Ограничение доступа к атрибутам и методам. Геттеры и сеттеры.

3 Программирование с графическим интерфейсом в Python

Графика с модулем tkinter в Python. Создание графического интерфейса. Импорт модуля tkinter. Менеджер размещения Grid. Текстовые поля и области. Виджеты и холст. Создание удобного пользовательского интерфейса

4 Основы процедурного программирования на Java

Структура программы. Алфавит языка, функция main, ее аргументы. Типы переменных в **Java**. Область видимости переменных. Целочисленные и дробные типы. Приведение типов. Ключевое слово final. Тип char, boolean.. Арифметика в **Java**. Операторы в **Java**. Условные управляющие конструкции в **Java**. Циклы в **Java**.

Ошибки компиляции: лексические, синтаксические, семантические. Массивы в **Java**. Многомерные массивы в **Java**. Демонстрация объектной природы массива. Типы **String** и **StringBuilder**. Задание методов в классе Main, рекурсия. Стек и его переполнение в рекурсии.

5 Знакомство с компилятором Java и документацией

Что такое **JDK**. Компилятор **Java**, вызов через javac. Байт-код, команда просмотра инструкций байт-кода javap. Что такое **JVM** (пара слов о языках **Clojure**, **Scala**, **Jython**, **jQuery**, **Kotlin**, **Nashorn**, **Groovy**), команда java. Параметр -classpath. **Javadocs**, пример. Знакомство с **IDE**: IntelliJ IDEA. Создание проекта и конфигурации. Сборка и исполнение программы в **IDE**. Дебаггер. Понятие стека вызовов.

6 Коллекции

Коллекции. Преимущества их использования. Интерфейс Collection. Утилитный класс Collections, его методы: shuffle, sort, unmodifiableSet, addAll. Конвертация коллекции в Array: toArray, правильный вариант использования этого метода. Утилитный класс Arrays, его методы: equals, asList. Возможность конструирования коллекций друг из друга. Интерфейс Iterable и итераторы. Доступ к итератору экземпляра коллекции. Методы hasNext, next, remove. Цикл **foreach**. Ограничения на тип-параметр коллекций: корректность реализации equals и hashCode. Интерфейс List, его свойства и методы. Использование метода subList для удаления и модификации подотрезка. Реализации интерфейса List: ArrayList и LinkedList, оценка времени операций, разница между ними. Интерфейс Queue, его свойства и методы, его наследник – интерфейс Deque, его реализации ArrayDeque и LinkedList. Интерфейс Set, его свойства и методы. Его реализации: HashSet и LinkedHashSet. Требования к типу-параметру в случае HashSet: корректные equals и hashCode, неизменность объектов с точки зрения hashCode в ходе работы с коллекцией. Сравнение HashSet и LinkedHashSet. Наследник Set'a: SortedSet, его свойства и методы. Его реализация TreeSet. Понятие компаратора. Интерфейс Map, его свойства и методы. Обход Map'ы: ключи, значения, пары ключ-значение.

7 Потоки данных

Потоки байт: InputStream, OutputStream. Методы read, skip, close InputStream'a. Методы write, flush, close OutputStream'a. Копирование из InputStream в OutputStream. Реализации InputStream и OutputStream. Интерфейсы FileInputStream, FileOutputStream, их создание с Files.newInputStream и Files.newOutputStream.. Классы ByteArrayInputStream, ByteArrayOutputStream. Запись в поток структурированных типизированных данных. Классы DataInputStream и DataOutputStream, оборачивание ими более низкоуровневого потока, примеры методов: writeInt, readInt. Классы DeflaterOutputStream и InflaterOutputStream. Потоки символов. Абстрактные классы Reader и Writer, их методы. Классы InputStreamReader и OutputStreamWriter, кодировка, класс StandardCharsets, класс Charset, методы forName, defaultCharset. Классы FileReader и FileWriter. Классы CharArrayReader, StringReader, CharArrayWriter и StringWriter. Класс BufferedReader, метод readLine. Классы newBufferedReader и newBufferedWriter. Классы PrintWriter и PrintStream. Класс Scanner, его параметры (разделитель, локаль). Сериализация и десериализация данных, маркерный интерфейс Serializable. Поле serialVersionUID, модификатор transient. Классы ObjectOutputStream и ObjectInputStream, методы readObject, writeObject.

8 Основы объектно-ориентированного программирования

Понятие класса, объекта. Инкапсуляция. Примеры. Поля класса, конструкторы (делегированные), методы класса, public / private / protected модификаторы, аннотации. **Immutable** классы, вложенные классы. Пакеты и модификаторы доступа: ключевое слово package, именование пакетов. Модификаторы доступа, инкапсуляция.

9 Наследование и абстракции

Концепция наследования, его возможности и применения. Реализация концепта наследования в **Java**: ключевое слово extends, конструктор наследника, ключевое слово super, переопределение ме-

тодов, аннотация `@Override`. Концепция полиморфизма. Реализация концепта динамического полиморфизма в **Java**. Абстрактные классы. Пример наследования: семейство геометрических фигур. Интерфейс, решение проблемы отсутствия множественного наследования). `default`-методы. Пример: `Comparator`.

10 Обработка ошибок и логирование

Исключение. Мотивировка их наличия. Вывод необработанных исключений в `stdout`. Демонстрация популярных исключений: `NullPointerException`, `ArrayIndexOutOfBoundsException`, `StringIndexOutOfBoundsException`, `FileNotFoundException`. Ошибки **JVM**, примеры: `OutOfMemoryError`, `NoClassDefFoundError`. Класс `Throwable`. Ключевое слово `throw`. Методы `getMessage`, `printStackTrace`, `getStackTrace`, `getCause`, `getSuppressed`. Группы исключений: `Error`, `Exception`, `RuntimeException`. Проверяемые исключения, ключевое слово `throws`. Создание собственного типа исключений. Обработка исключений, `try / catch`, несколько блоков `catch`, несколько типов исключений в одном блоке `catch`, блок `finally`, повторный `throw` из `try / catch`. Логирование, пакет `java.util.logging`, класс `Logger`. Прикрепление класса логирования к классу. Методы `log`, `warning`, `setLevel`. Использование динамических данных для логирования, передача форматной строки в метод `log`. Класс `java.util.logging.Handler`, его имплементации: `ConsoleHandler`, `FileHandler`, `SocketHandler`. Метод `addHandler`. Класс `java.util.logging.Formatter`, его имплементации: `SimpleFormatter`, `XMLFormatter`. Управление логированием с использованием конфигурационного файла.

11 Дженирики. Функциональные интерфейсы

Мотивация существования параметризованных типов (**Generic**'ов). Ограничения **Generic**'ов. Параметризация класса **generic**'ом, создание поля-**generica**, метода, использующего **generic**. Невозможность использования **generic**'а в статическом методе. Параметризация метода **generic**'ом. Ключевое слово `extends`. Синтаксическая конструкция `G<?>` (эквивалентна `G<? extends Object>`). Пример реализации класса с использованием дженериков. Инстанцирование **generic**-объектов. Лямбда-выражения. Синтаксические правила записи. Примеры использования. Интерфейсы. Актуальность их использования в коде. Пример.

Функциональный интерфейс, аннотация `@FunctionalInterface`, интерфейс `Runnable`. Инстанцирование функциональных интерфейсов в виде именнованного, либо анонимного класса, лямбда-выражения, либо ссылки на метод. Пакет `java.util.function`. Группы функциональных интерфейсов: `Consumer`'ы, `Supplier`'ы, `Predicate`'ы, `Function`'ы, `Operator`'ы. Полезные методы встроенных функциональных интерфейсов: методы `negate` и `add` у `Predicate`'ов, методы `addThen` у `Consumer`'ов, методы `addThen` и `compose` у `Function` (и их различие), метод `comparing` у `Comparator`'ов.

12 Стримы

Stream API. Интерфейс `Stream`. Отличия стрима от итератора. Отличия стрима от коллекции. Примеры получения стрима из коллекции, `Reader`'а, директории, строки, массива, списка элементов. Генерирование `Stream`'а из `Supplier`'а. Получение `Stream`'а из `range`'а. Получение `Stream`'а итерированием по функции. Конкатенация `Stream`'ов, получение пустого `Stream`'а. Закрытие `Stream`'а (и необходимость этой операции). Операции: Преобразования: `filter`, `limit`, `map`, `mapToObj`, `flatMapToInt`, `distinct`, `sorted`, `skip`, `peek` (передача `System.out::println`), терминальные операции: `forEach`, `findFirst`, `findAny`, `allMatch`, `anyMatch`, `noneMatch`, `min`, `max`, `count`, `sum`, `reduce`. Остальные: `iterate`, `toArray`.

13 Рефлексия и аннотации

Класс `Class`. Три способа получения экземпляра класса `Class`. Методы `Class`, их отличия. Класс `Modifier` и его основные статические методы. Класс `Constructor`, его основные методы и метод `newInstance`. Аннотации: мотивация, примеры аннотаций: `Deprecated`, `Override`, `SuppressWarnings`. Создание собственной аннотации, аннотации `Target`, `Retention`, `Inherited`, `Documented`. Все возможные значения `target` и `retention`. Использование рефлексии для создания обработчика аннотаций. Методы `Class::isAnnotationPresent`, `Class::getAnnotations`, `Class::getAnnotation`, `Class::isAnnotation`

14 Unit-тестирование. Java-doc

Зачем писать модульные тесты. Библиотека `JUnit` и `TestNG`. Их сравнение. `Mock Object`. Назначение моков. `Javadoc`: комментарии, документация, теги.

4.3 Лабораторные работы

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
1	1	Методы в Python	2
2	1	Итераторы, генераторы, декораторы функций в Python	4
3	1	Функциональное программирование в Python	4
4	2	Классы и объекты в Python	4
5	2	Наследование в Python	6
6	2	Абстрактные методы в Python	4
7	2	Полиморфизм в Python	4
8	3	Знакомство с библиотеками Python для графического интерфейса	4
9	3	Создание приложений с графическим интерфейсом	6
10	4	Управляющие конструкции Java	1
11	5	Сложные типы данных. Строки. Методы работы со строками	1
12	6	Коллекции	1
13	7	Методы. Потоки данных	1
14	8	Классы и объекты	2
15	9	Наследование.	2
16	10	Полиморфизм. Абстрактные методы	2
17	11	Обработка ошибок. Логирование	2
18	12	Интерфейсы. Функциональные интерфейсы	2
19	13	Стримы	1
20	14	Рефлексия и аннотации	1
21	14	Unit-тестирование. Java-doc	1
		ИТОГО	50

4.4 Курсовая работа (5 семестр)

Предусмотрена курсовая работа в 5 семестре. Тема курсовой работы связана с проектированием и разработкой приложения для работы с базой данных некоторой предметной области.

Примерные темы курсовых работ:

- 1) Разработка информационной системы данных о спортивных мероприятиях
- 2) Разработка системы информационной поддержки работы аэропорта
- 3) Разработка информационной системы «Туристическое агентство»
- 4) Разработка системы информационной поддержки метеорологической станции
- 5) Разработка информационной системы «Ветеринарная клиника»
- 6) Разработка электронного журнала учета успеваемости
- 7) Разработка информационной системы «Киносправочник»
- 8) Разработка системы информационной поддержки интернет-магазина
- 9) Разработка системы информационной поддержки книжного магазина
- 10) Разработка информационной системы «Литературный справочник»

5 Учебно-методическое обеспечение дисциплины

5.1 Основная литература

1. Крыжановская, Ю. А. Основы JAVA : учебное пособие / Ю. А. Крыжановская, В. Г. Ляликова, М. М. Безрядин. — Воронеж : ВГУ, 2020. — 96 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/433010> (дата обращения: 07.04.2026). — Режим доступа: для авториз. пользователей.
2. Седжвик, Р. Фундаментальные алгоритмы на JAVA: пер. с англ. / Р. Седжвик. - 3-я ред. - М. ; СПб. ; Киев : DiaSoft, 2003.. - ISBN 966-7992-22-5. Ч. 1-4 : Анализ. Структуры данных. Сортировка. Поиск. - , 2003. - 688 с. : ил. - Предм. указ.: с. 676-680. - ISBN 966-8035-28-3.

5.2 Дополнительная литература

1. Хорстманн, К. С. Java 2 [Текст] : Core Java / К. С. Хорстманн, Г. Корнелл. - 8-е изд. - М. : Вильямс, 2009. - (Библиотека профессионала).. - ISBN 978-5-8459-1482-8 Т. 1 : Основы. - , 2009. - 814 с. : ил. - Прил.: с. 805-806. - Предм. указ.: с. 807-813. - ISBN 978-5-8459-1378-4. - ISBN 978-0-1323-5476-9.
2. Дейтел, Х. М. Как программировать на Java [Текст] / Х. М. Дейтел, П. Д. Дейтел. - М. : Бин-ом-Пресс. Кн. 1 : Основы программирования. - 4-е изд. - 2003. - 848 с.: ил. - ISBN 5-9518-0015-3.
3. Шилдт, Г. Swing = Swing: A Beginner's Guide [Текст] : рук. для начинающих: пер. с англ. / Г. Шилдт. - М. : Вильямс, 2007. - 698 с. - (Основы основ - проще простого!). - Парал. тит. л. англ. - Прил.: с. 627-688. - Предм. указ.: с. 689-697. - ISBN 978-5-8459-1162-9. - ISBN 0-07-226314-8.

5.3 Периодические издания

Журнал «Прикладная информатика»

5.4 Интернет-ресурсы

1. <http://www.citforum.ru/> - портал аналитических и научных статей в области информационных технологий
2. <http://www.rsdn.ru> - сайт Российской сети разработчиков ПО, содержит статьи по современным средствам программирования.

<http://www.intuit.ru> – сайт Интернет-университета информационных технологий, представляет учебные курсы по разным областям ИТ.

5.5 Программное обеспечение, профессиональные базы данных и информационные справочные системы современных информационных технологий

- 1 Операционная система РЕД ОС для рабочих станций, имеется лицензия, входит в реестр отечественного ПО. – Режим доступа: <https://redos.red-soft.ru/>
- 2 LibreOffice – свободно распространяемый офисный пакет программ, включающий в себя текстовый и табличный редакторы, редактор презентаций и другие офисные приложения. – Режим доступа: <https://ru.libreoffice.org/>
- 3 Программа для просмотра сайтов Яндекс.Браузер, свободно распространяемая, входит в реестр отечественного ПО. – Режим доступа: <https://yandex.ru/>
- 4 Программная система для организации видео-конференц-связи DION. – Режим доступа: <https://diongo.ru/>
- 5 Университетская платформа электронного обучения «Электронные курсы ОГУ в системе обучения Moodle». – Режим доступа: <http://moodle.osu.ru>
- 6 Автоматизированная интерактивная система сетевого тестирования (АИССТ). – Режим доступа: <https://aist.osu.ru/cgi-bin/auth.cgi>

7 Университетская платформа для сопровождения процедуры проведения экзаменационных испытаний с использованием дистанционных образовательных технологий. – Режим доступа: <https://exam.osu.ru/>

8 Система программирования Python, свободно распространяемая по лицензии PSFL.

9 Интегрированная среда разработки ПО NetBeans, свободно распространяемая по лицензии Apache.

10 Интегрированная среда разработки ПО Visual Studio Code, свободно распространяемая по лицензии MIT.

11 Средства для разработки JetBrains All Products Pack, бесплатно лицензируемая для образовательного учреждения (включает C++, Java, C#, PHP, Python...).

6 Материально-техническое обеспечение дисциплины

Учебные аудитории для проведения занятий лекционного типа, семинарского типа, для проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Аудитории оснащены комплектами ученической мебели, техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения лабораторных занятий компьютерный класс, оснащенный компьютерной техникой, подключенной к сети "Интернет" с обеспечением доступа в электронную информационно-образовательную среду ОГУ.