Минобрнауки России

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

Кафедра математики и цифровых технологий

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«ФДТ.1 Объектно-ориентированные языки и системы»

Уровень высшего образования

МАГИСТРАТУРА

Направление подготовки
09.04.02 Информационные системы и технологии
(код и наименование направления подготовки)

Искусственный интеллект в промышленности
(наименование направленности (профиля) образовательной программы)

Квалификация
Магистр

Форма обучения
Очная

Год набора 2026

Рабочая программа дисциплины «*ФДТ.1 Объектно-ориентированные языки и системы*» рассмотрена и утверждена на заседании кафедры

Кафедра математики и цифровых технологий
наименование кафедры

протокол № 7 от "19" марта 2026 г.

Заведующий кафедрой
Кафедра математики и цифровых технологий _____ А.Е. Шухман
наименование кафедры *подпись* *расшифровка подписи*

Исполнители:
Старший преподаватель кафедры МЦТ _____ И.В. Минина
должность *подпись* *расшифровка подписи*

_____ _____ _____
должность *подпись* *расшифровка подписи*

СОГЛАСОВАНО:
Председатель методической комиссии по направлению подготовки
09.04.02 Информационные системы и технологии _____ И.П. Болодурина
код наименование *личная подпись* *расшифровка подписи*

Научный руководитель магистерской программы _____ И.П. Болодурина
личная подпись *расшифровка подписи*

Заведующий отделом формирования фонда и научной обработки документов
_____ С.А. Биктимирова
личная подпись *расшифровка подписи*

Уполномоченный по качеству института _____ С.Н. Морозова
личная подпись *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 объектно-ориентированных языков программирования (ООЯ);

– изучить основные свойства, средства и утилиты платформ ООЯ, возможности данных языков;

– научиться разрабатывать иерархию классов на ООЯ 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 является факультативной(ым)

Пререквизиты дисциплины: *Отсутствуют*

Постреквизиты дисциплины: *Отсутствуют*

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
ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2-В-1 Разрабатывает и реализует алгоритмы и программы обработки различных структур данных ОПК-2-В-2 Оценивает эффективность алгоритмов обработки структур данных ОПК-2-В-3 Разрабатывает оригинальные алгоритмы и программное обеспечение с использованием современных интеллектуальных технологий ОПК-2-В-4 Создает алгоритмы реализации моделей и программные средства для поддержки принятия решений	**Знать:** – основные понятия, принципы и модели ООП пользовательские типы данных – классы, объект, объявление и реализацию класса, данные, методы; доступ, наследование свойств, виртуальные функции; **Уметь:** – определять классы, создавать объекты, разрабатывать и использовать конструкторы класса; создавать программы, используя библиотеки классов; **Владеть:** – навыками объектно-ориентированного программирования на языке Java.

4 Структура и содержание дисциплины

4.1 Структура дисциплины

Общая трудоемкость дисциплины составляет 3 зачетные единицы (108 академических часов).

Вид работы	Трудоемкость, академических часов	
	1 семестр	всего
Общая трудоёмкость	**108**	**108**
Контактная работа:	**16,25**	**16,25**
Практические занятия (ПЗ)	16	16
Промежуточная аттестация (зачет, экзамен)	0,25	0,25
Самостоятельная работа: *- самоподготовка (проработка и повторение материала учебников и учебных пособий)* *- изучение разделов курса в системе электронного обучения;* *- подготовка к практическим занятиям;* *- подготовка к рубежному контролю*	**91,75**	**91,75**
Вид итогового контроля (зачет, экзамен, дифференцированный зачет)	**зачет**	

Разделы дисциплины, изучаемые в 1 семестре

№ раздела	Наименование разделов	Количество часов				
		всего	аудиторная работа			внеауд. работа
			Л	ПЗ	ЛР	
1	Основы Java Core	12		2		10
2	Основы объектно-ориентированного программирования. Принципы ООП.	12		2		10
3	Наследование и полиморфизм	14		2		12
4	Внутренние и вложенные классы. Абстрактные классы и интерфейсы, функциональные интерфейсы	14		2		12
5	Обработка ошибок и логирование. Unit-тестирование. Java-doc	14		2		12
6	Дженерики	14		2		12
7	Stream API	14		2		12
8	Многопоточность	14		2		12
	Итого:	108		16		92
	Всего:	108		16		92

4.2 Содержание разделов дисциплины

1. Основы Java Core

Примитивные типы данных, ссылочные типы, литералы. Классы обертки. Операторы: арифметические, логические, сравнения, тернарный. Управляющие конструкции. Класс String. Способы создания строк. Пул строк, интернирование. Классы StringBuilder и StringBuffer. Фреймворк Collections. Иерархия (Iterable, Collection, List, Set, Queue). Классы-реализации: ArrayList, LinkedList, HashSet, TreeSet, LinkedHashSet. Интерфейс Map: HashMap, TreeMap, LinkedHashMap. Определение и вызов методов. Синтаксис метода: модификаторы, тип возвращаемого значения, имя, параметры, тело. Сигнатура метода. Исключения: иерархия. Проверяемы (checked) и непроверяемые (unchecked)

исключения. Конструкция try-catch-finally. Генерация исключений: throw, throws. Создание собственных исключений.

2. Основы объектно-ориентированного программирования. Принципы ООП

Введение в ООП. Понятия класса, объекта, поля, метода. Поля экземпляра и статические поля. Константы (final). Конструкторы: по умолчанию, параметризованные, цепочка вызовов this(). Методы: возвращаемые значения, параметры, перегрузка методов. Инкапсуляция: модификаторы доступа (private, package-private, protected, public). Геттеры и сеттеры. Валидация данных. Статические методы и поля. Класс Math, утилитарные классы. Объекты как параметры и возвращаемые значения методов. Классы-обёртки (wrapper classes), автобоксинг/анбоксинг.

3. Наследование и полиморфизм

Наследование. Ключевое слово extends, переопределение методов (@Override). Вызов родительских конструкторов super(). Защищённые (protected) члены класса. Полиморфизм: Виртуальные методы. Демонстрация позднего связывания.Абстрактные классы и абстрактные методы. Ключевое слово final (класс, метод, поле). Класс Object и его методы: toString(), equals(), hashCode(). Правила переопределения equals() и hashCode().

4. Внутренние и вложенные классы. Абстрактные классы и интерфейсы. Функциональные интерфейсы

Определение внутренних и вложенных классов в Java. Общая классификация: статические вложенные классы и нестатические внутренние классы. Отличия между вложенными и внутренними классами. Статические вложенные классы, их синтаксис объявления и создания. Особенности доступа к членам внешнего класса и правила создания экземпляров. Нестатические внутренние классы. Неявная связь с экземпляром внешнего класса. Доступ к полям и методам внешнего класса (в том числе к приватным). Анонимные классы. Создание анонимных классов на основе интерфейсов и абстрактных классов. Расширение конкретных классов с помощью анонимных классов. Доступ к переменным внутри анонимного класса.

Особенности абстрактных классов. Отличие абстрактного класса от конкретного (обычного) класса. Абстрактные методы (без тела, только сигнатура). Наследование от абстрактного класса. Интерфейсы. Понятие интерфейса как контракта. Отличие от абстрактного класса. Реализация интерфейсов. Наследование интерфейсов. Проблема ромба. Конфликт при наследовании двух интерфейсов с одинаковыми default методами. Правила разрешения конфликтов. Сравнение и выбор: Abstract Class vs Interface.

Определение функционального интерфейса (interface с ровно одним абстрактным методом). Аннотация @FunctionalInterface (назначение, правила проверки). Чем отличается от обычного интерфейса. Лямбда-выражения. встроенные функциональные интерфейсы. Специализированные встроенные функциональные интерфейсы для примитивов. Создание своих функциональных интерфейсов.

5. Обработка ошибок и логирование. Unit-тестирование. Java-doc

Создание и выбрасывание исключений. Операторы throw. Throws в сигнатуре метода. Создание собственных исключений. Checked и unchecked исключения. Механизм проброса исключений (propagation). Стек вызовов (call stack). Непойманные исключения. Конструкция try-with-resources. Порядок закрытия ресурсов (обратный созданию).

Введение в логирование. Уровни логирования. Стандартные фреймворки логирования. Подключение зависимостей (Maven/Gradle). Получение логгера. Параметризованные сообщения. Конфигурация логирования. Аппендеры (назначение вывода).

Понятие Unit-теста, Unit-тестирования. Отличие от интеграционного, E2E, регрессионного тестов. Принципы хороших тестов. Жизненный цикл теста. Фреймворк JUnit. Структура тестового класса. Аннотации. Жизненный цикл тестового экземпляра. Ассерты (проверки). Параметризованные тесты. Вложенные тесты. Mockito (изоляция зависимостей). Когда нужны mock-объекты. Основные возможности Mockito.

Зачем нужна документация. Проблема недокументированного кода. Что такое JavaDoc. Утилита javadoc из JDK. Извлечение структурированных комментариев в HTML. Виды комментариев в Java. Автоматическая генерация сайта документации. Синтаксис базовых тегов.

6. Дженерики (обобщения)

Зачем нужны обобщения .Проблема сырых типов (до Java 5). Что дают Generics. Базовый синтаксис Generics. Generic-классы, интерфейсы, методы и конструкторы. Ограничения на применение дженериков. Подстановочные знаки (Wildcards) в дженериках. Стирание типов.

7. Stream API

Что такое Stream API (Java 8+). Создание стримов. Источник (source) для Stream API: коллекция, массив, файл, генератор. Генерация бесконечных стримов. Промежуточные операции. Терминальная операция (terminal). Конвейер (pipeline). Сравнение с коллекциями. Collectors: группировка, разделение,строковые операции, кастомный коллектор. Примитивные стримы. Параллельные стримы. Ленивость и оптимизации.

8. Многопоточность

Определение процесса и потока (thread). Преимущества многопоточности. Проблемы многопоточности. Модель памяти Java. Создание и запуск потоков. Жизненный цикл потока. Базовое управление потоками. Синхронизация и разделяемые ресурсы. Атомарные операции и потокобезопасные контейнеры. Пул потоков. Виртуальные потоки. Тестирование многопоточного кода.

4.3 Практические занятия (семинары)

№ занятия	№ раздела	Тема	Кол-во часов
1	1	Основы Java Core	2
2	2	Основы объектно-ориентированного программирования. Принципы ООП.	2
3	3	Наследование и полиморфизм	2
4	4	Внутренние и вложенные классы. Абстрактные классы и интерфейсы, функциональные интерфейсы	2
5	5	Обработка ошибок и логирование. Unit-тестирование. Java-doc	2
6	6	Дженерики	2
7	7	Stream API	2
8	8	Многопоточность	2
		Итого:	16

5 Учебно-методическое обеспечение дисциплины

5.1 Основная литература

1. Крыжановская, Ю. А. Основы JAVA : учебное пособие / Ю. А. Крыжановская, В. Г. Лялико-ва, 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. : Бином-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. - 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

1. Информационные технологии : журнал. - М. : Агентство "Роспечать".
2. Программные продукты и системы : журнал. - М. : Агентство "Роспечать".

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