



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГЕОДЕЗИИ И КАРТОГРАФИИ
(МИИГАиК)

Факультет геоинформатики и информационной безопасности
Кафедра информационно-измерительных систем



**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5387FD2AF5B17FED816FDE9A1D4973B1
Владелец: Степанченко Алексей Леонидович
Действителен: с 21.07.2025 по 14.10.2026

«УТВЕРЖДАЮ»

Проректор
А. Л. Степанченко
16 февраля 2024 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
«Б1.О.13 ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»**

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.02 Информационные системы и технологии

Направленность(профиль) подготовки: Информационные системы и технологии

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2024

Срок получения образования: 4 года

Объем:
в зачетных единицах: 4 з.е.
в академических часах: 144 ак.ч.

Москва, 2024

Разработчики:

Преподаватель кафедры Информационно-измерительных систем
Лазуренко Н. С.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии, утвержденного приказом Минобрнауки от 19.09.2017 № 926

Согласование и утверждение

№	Подразделение	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1	факультет Геоинформатики и информационной безопасности	Председатель ученого совета факультета	Матерухин А. В.	Рассмотрено	14.12.2023, № 7-23

Содержание

1. Цель и задачи освоения дисциплины (модуля)
2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы
3. Место дисциплины в структуре ОП
4. Объем дисциплины и виды учебной работы
5. Содержание дисциплины
 - 5.1. Разделы, темы дисциплины и виды занятий
 - 5.2. Содержание разделов, тем дисциплины и формы текущего контроля
6. Оценочные материалы текущего контроля
7. Оценочные материалы промежуточной аттестации
8. Порядок проведения промежуточной аттестации
9. Материально-техническое и учебно-методическое обеспечение дисциплины
 - 9.1. Перечень основной и дополнительной учебной литературы
 - 9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся
 - 9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине
 - 9.4. Оборудование и технические средства обучения
10. Методические указания по освоению дисциплины (модуля)

1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - формирование у студентов базовых компетенций объектно-ориентированного программирования на языке Java. Студенты научатся проектировать, реализовывать и документировать программы с использованием принципов ООП, включая инкапсуляцию, наследование и полиморфизм.

Задачи изучения дисциплины:

- освоение основ синтаксиса и семантики языка Java;
- понимание и применение основных концепций объектно-ориентированного проектирования (ООП);
- практическое освоение разработки программного обеспечения с использованием инструментов и библиотек Java SE.

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

Компетенции, индикаторы и результаты обучения

ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

ОПК-6.1 Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем

Знать:

ОПК-6.1/Зн1 методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий

Уметь:

ОПК-6.1/Ум1 применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий

Владеть:

ОПК-6.1/Нв1 навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач

ОПК-6.2 Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий

Знать:

ОПК-6.2/Зн1 методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий

Уметь:

ОПК-6.2/Ум1 применять методы и средства анализа функциональных требований к программному обеспечению

Владеть:

ОПК-6.2/Нв1 навыками анализа функциональных требований к программному обеспечению

3. Место дисциплины в структуре ОП

Дисциплина (модуль) Б1.О.13 «Объектно-ориентированное программирование» относится к обязательной части образовательной программы и изучается в семестре(ах): 4.

В процессе изучения дисциплины студент готовится к решению типов задач профессиональной деятельности, предусмотренных ФГОС ВО и образовательной программой.

4. Объем дисциплины (модуля) и виды учебной работы

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Практические занятия (часы)	Самостоятельная работа (часы)	Промежуточная аттестация (часы)
4 семестр	144	4	34	110	Зачет с оценкой
Всего	144	4	34	110	

5. Содержание дисциплины (модуля)

5.1. Разделы, темы дисциплины и виды занятий

(часы промежуточной аттестации не указываются)

Наименование раздела, темы	Всего	Практические занятия	Самостоятельная работа	Планируемые результаты обучения, соотнесенные с результатами освоения программы
Раздел 1. Основы языка Java	32	8	24	ОПК-6.1
Тема 1.1. Основные конструкции языка Java (переменные, типы данных, операторы)	8	2	6	
Тема 1.2. Управление потоком исполнения (условия, циклы)	8	2	6	
Тема 1.3. Массивы и строки	8	2	6	
Тема 1.4. Исключительные ситуации (try-catch-finally)	8	2	6	
Раздел 2. Классы и объекты	32	8	24	ОПК-6.1
Тема 2.1. Создание классов и объектов	8	2	6	
Тема 2.2. Инкапсуляция и доступ к данным	8	2	6	
Тема 2.3. Конструкторы и деструкторы	8	2	6	
Тема 2.4. Статические члены класса	8	2	6	

Раздел 3. Наследование и интерфейсы	32	8	24	ОПК-6.2
Тема 3.1. Принцип наследования	8	2	6	
Тема 3.2. Полиморфизм и перегрузка методов	8	2	6	
Тема 3.3. Интерфейсы и абстрактные классы	8	2	6	
Тема 3.4. Пакеты и видимость членов класса	8	2	6	
Раздел 4. Коллекции и обобщения	22	6	16	ОПК-6.2
Тема 4.1. Обобщённые коллекции (List, Set, Map)	8	2	6	
Тема 4.2. Обработка коллекций (stream API)	8	2	6	
Тема 4.3. Лямбда-выражения и функциональные интерфейсы	6	2	4	
Раздел 5. Работа с файлами и ввод-вывод	26	4	22	ОПК-6.1 ОПК-6.2
Тема 5.1. Файловые потоки (FileInputStream, FileOutputStream)	8	2	6	
Тема 5.2. Форматы сериализации (XML, JSON)	8	2	6	
Тема 5.3. Работа с сетью (Socket, HTTP)	10		10	
Итого	144	34	110	

5.2. Содержание разделов, тем дисциплин и формы текущего контроля

Раздел 1. Основы языка Java

Тема 1.1. Основные конструкции языка Java (переменные, типы данных, операторы)

Определение и использование переменных для хранения данных; правила именования и область видимости переменных. Обзор примитивных типов данных (int, float, char, boolean) и ссылочных типов; различия между ними и применение в программировании. Изучение арифметических, логических и условных операторов; использование операторов для выполнения различных операций над данными.

Тема 1.2. Управление потоком исполнения (условия, циклы)

Обзор операторов if, else, switch; применение для принятия решений в программе на основе условий. Изучение циклов for, while, do-while; применение для повторного выполнения блока кода, управление итерациями и условиями выхода из цикла.

Тема 1.3. Массивы и строки

Определение массивов как коллекций однотипных элементов; создание, инициализация и доступ к элементам массива; методы работы с массивами. Изучение строк как объектов в Java; методы класса String для манипуляции строками (сравнение, конкатенация, извлечение подстрок); работа с неизменяемыми строками.

Тема 1.4. Исключительные ситуации (try-catch-finally)

Понимание концепции исключений в Java; использование блоков try, catch и finally для обработки ошибок и предотвращения сбоев программы. Изучение возможности создания пользовательских исключений для обработки специфических ошибок в приложении; применение принципов хорошего дизайна при обработке исключений.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 2. Классы и объекты

Тема 2.1. Создание классов и объектов

Определение классов как шаблонов для создания объектов; структура класса, включая поля (атрибуты) и методы (функции). Понимание объектов как экземпляров классов; создание объектов с использованием оператора new; доступ к полям и методам объектов.

Тема 2.2. Инкапсуляция и доступ к данным

Принцип скрытия внутренней реализации класса; использование модификаторов доступа (private, protected, public) для ограничения доступа к полям и методам. Создание методов доступа (геттеров и сеттеров) для управления доступом к приватным полям класса; применение этих методов для обеспечения контроля над изменением состояния объекта.

Тема 2.3. Конструкторы и деструкторы

Понимание конструкторов как специальных методов, вызываемых при создании объекта; создание перегруженных конструкторов для инициализации объектов с различными параметрами. Обзор концепции деструкторов в Java (хотя Java не поддерживает их в традиционном смысле); использование метода finalize() для освобождения ресурсов перед сборкой мусора, хотя это не рекомендуется.

Тема 2.4. Статические члены класса

Определение статических членов класса, которые принадлежат самому классу, а не конкретным объектам; использование статических полей для хранения общих данных и статических методов для выполнения операций без создания экземпляра класса. Примеры использования статических членов в реальных приложениях; понимание, когда целесообразно использовать статические элементы, а когда лучше работать с экземплярами классов.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 3. Наследование и интерфейсы

Тема 3.1. Принцип наследования

Определение наследования как механизма, позволяющего одному классу (дочернему) унаследовать свойства и методы другого класса (родительского); ключевое слово `extends` для указания наследования. Понимание возможности переопределения методов родительского класса в дочернем; использование аннотации `@Override` для обозначения переопределенных методов. Обзор отсутствия множественного наследования классов в Java; использование интерфейсов для достижения аналогичного эффекта.

Тема 3.2. Полиморфизм и перегрузка методов

Определение полиморфизма как способности объектов разных классов реагировать на одинаковые сообщения (методы); различие между статическим (перегрузка) и динамическим (переопределение) полиморфизмом. Понимание перегрузки методов как возможности определения нескольких методов с одинаковым именем, но с разными параметрами; примеры использования перегрузки для улучшения читаемости и удобства кода.

Тема 3.3. Интерфейсы и абстрактные классы

Определение интерфейсов как контрактов, которые классы могут реализовывать; ключевое слово `implements` для реализации интерфейса; возможность создания методов по умолчанию в интерфейсах (с Java 8). Понимание абстрактных классов как классов, которые не могут быть инстанцированы и могут содержать абстрактные методы (без реализации); использование абстрактных классов для создания базовых классов с общими характеристиками для дочерних классов.

Тема 3.4. Пакеты и видимость членов класса

Определение пакетов как механизмов организации классов и интерфейсов в Java; создание пользовательских пакетов с использованием ключевого слова `package`. Обзор модификаторов доступа (`public`, `protected`, `private`, и пакетная видимость) и их влияние на доступ к классам, методам и полям; понимание, как выбор модификатора доступа влияет на инкапсуляцию и безопасность кода.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 4. Коллекции и обобщения

Тема 4.1. Обобщённые коллекции (*List, Set, Map*)

Введение в обобщённые коллекции как способ хранения объектов разных типов с использованием параметров типа; преимущества обобщений, включая безопасность типов и устранение необходимости в приведении типов. Определение интерфейса `List` как упорядоченной коллекции, допускающей дубликаты; основные реализации (`ArrayList`, `LinkedList`); методы для добавления, удаления и поиска элементов. Определение интерфейса `Set` как коллекции, не допускающей дубликатов; основные реализации (`HashSet`, `TreeSet`); особенности работы с уникальными элементами и порядок хранения. Определение интерфейса `Map` как коллекции пар "ключ-значение"; основные реализации (`HashMap`, `TreeMap`, `LinkedHashMap`); методы для добавления, удаления и поиска значений по ключу.

Тема 4.2. Обработка коллекций (stream API)

Введение в Stream API как способ обработки последовательностей данных с использованием функционального подхода; создание потоков из коллекций и массивов. Обзор промежуточных (например, filter, map, sorted) и терминальных (например, collect, forEach, reduce) операций; понимание ленивой и жадной обработки данных. Возможности параллельной обработки данных с использованием метода parallelStream(); преимущества и недостатки параллельного выполнения.

Тема 4.3. Лямбда-выражения и функциональные интерфейсы

Определение лямбда-выражений как краткого способа описания анонимных функций; синтаксис лямбда-выражений и их использование для упрощения кода. Понятие функциональных интерфейсов как интерфейсов, содержащих ровно один абстрактный метод; примеры стандартных функциональных интерфейсов (Runnable, Callable, Consumer, Function, Predicate). Использование лямбда-выражений в сочетании с коллекциями и Stream API для более лаконичного и выразительного кода; примеры применения в реальных сценариях.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 5. Работа с файлами и ввод-вывод

Тема 5.1. Файловые потоки (FileInputStream, FileOutputStream)

Введение в концепцию файловых потоков как способа чтения и записи данных в файлы; различие между байтовыми и символьными потоками. Описание класса FileInputStream для чтения байтов из файла; методы для открытия файла, чтения данных и закрытия потока; обработка исключений при работе с файлами. Описание класса FileOutputStream для записи байтов в файл; методы для создания файла, записи данных и закрытия потока; использование буферизации для повышения производительности. Примеры кода для чтения из текстовых и бинарных файлов, а также записи данных в файлы, включая обработку ошибок.

Тема 5.2. Форматы сериализации (XML, JSON)

Определение сериализации как процесса преобразования объектов в формат, пригодный для хранения или передачи; преимущества сериализации для сохранения состояния объектов. Описание формата XML как текстового формата для хранения структурированных данных; использование библиотек (например, JAXB) для сериализации и десериализации объектов в XML и обратно. Описание формата JSON как легковесного текстового формата для обмена данными; использование библиотек (например, Jackson или Gson) для сериализации и десериализации объектов в JSON и обратно. Сравнение XML и JSON по критериям читаемости, размера, производительности и удобства использования; выбор подходящего формата в зависимости от требований проекта.

Текущий контроль

Вид (форма) контроля
Практическая работа

Тема 5.3. Работа с сетью (Socket, HTTP)

Введение в основы сетевого программирования в Java; понятие клиент-серверной модели и протоколов передачи данных. Описание класса Socket для создания клиентских сокетов; методы для подключения к серверу, отправки и получения данных; работа с потоками ввода/вывода через сокет. Описание класса ServerSocket для создания серверных сокетов; методы для ожидания входящих соединений и обработки запросов от клиентов. Обзор протокола HTTP как основного протокола для передачи данных в интернете; использование библиотеки HttpURLConnection для выполнения HTTP-запросов (GET, POST) и обработки ответов. Примеры создания простых клиент-серверных приложений, работающих через сокет и HTTP, включая обработку ошибок и многопоточность.

Текущий контроль

Вид (форма) контроля
Тестирование

6. Порядок проведения промежуточной аттестации

Порядок проведения промежуточной аттестации регламентируется локальными нормативными актами Университета:

- Положением о формах, периодичности, порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по основным образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.6 «Организация и проведение промежуточной аттестации».
- Положением о порядке осуществления образовательного процесса, проведения текущего контроля успеваемости и промежуточной аттестации обучающихся с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.8 «Проведение промежуточной аттестации с применением ЭО и ДОТ».

Промежуточная аттестация: Зачет с оценкой, 4 семестр.

Зачет оценивается по пятибалльной шкале:

- зачтено с оценкой «отлично» выставляется, если обучающийся имеет глубокие, исчерпывающие знания в объеме освоенной программы; знание основной (обязательной) литературы; дает правильные и уверенные ответы, свидетельствующие о наличии твердых знаний и навыков в использовании технических средств; полное, четкое, грамотное и логически стройное изложение материала; демонстрирует свободное применение теоретических знаний при анализе практических вопросов;
- зачтено с оценкой «хорошо» выставляется обучающемуся в случае соблюдения тех же требований, но наличии в ответе обучающегося по некоторым перечисленным показателям недостатков принципиального характера, что вызывает замечания или поправки преподавателя;
- зачтено с оценкой «удовлетворительно» выставляется обучающемуся в случае соблюдения тех же требований, но в случае наличия в ответе ошибок, необходимости существенной помощи в виде поправок и наводящих вопросов преподавателя;
- не зачтено «не удовлетворительно» – обучающийся не смог ответить на 2/3 вопросов; не справился с практическим заданием или выполнено менее 50 % практических заданий.

7. Оценочные материалы текущего контроля

Раздел 1. Основы языка Java

Контролируемые ИДК: ОПК-6.1

Тема 1.4. Исключительные ситуации (try-catch-finally)

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Что означает аббревиатура ООР?

1. Object-Oriented Programming
2. Open Online Platform
3. Overloading Operators Protocol
4. Operating Optimization Process

2. Какой метод автоматически выполняется при создании нового объекта?

1. main()
2. constructor()
3. finalize()
4. static init()

3. Для чего служит ключевое слово super?

1. Обращается к родительскому классу
2. Создает новый экземпляр текущего класса
3. Увеличивает приоритет потока
4. Используется для инициализации массивов

4. Какой класс хранит элементы без дубликатов?

1. List
2. Array
3. Set
4. Queue

5. Какой интерфейс позволяет сортировать коллекцию?

1. Comparable
2. Collection
3. Serializable
4. Cloneable

6. Почему рекомендуется скрывать внутренние данные класса?

1. Это повышает производительность
2. Это улучшает читаемость кода
3. Это упрощает работу компилятора
4. Это способствует инкапсуляции

Раздел 2. Классы и объекты

Контролируемые ИДК: ОПК-6.1

Тема 2.4. Статические члены класса

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Где хранятся локальные переменные и аргументы методов?

1. В стеке
2. В куче
3. В регистрах процессора
4. В глобальной области видимости

2. Чему равен результат выражения (byte) (1 << 3)?

1. 8
2. 4
3. 1
4. 0

3. Как создать ссылку на родительский класс внутри подкласса?

1. super
2. parent

3. this
4. base

4. Каково назначение аннотации @Override?

1. Проверяет правильность сигнатуры переопределенного метода
 2. Изменяет порядок вызова методов
 3. Позволяет создавать анонимные классы
 4. Используется для сокрытия полей класса
5. Если один класс расширяется другим, какой термин используется?
1. Преобразование
 2. Переход
 3. Наследование
 4. Агрегация
6. Сколько уровней вложенности могут иметь абстрактные классы?
1. Только один уровень
 2. Два уровня
 3. Любое количество уровней
 4. Ни одного уровня

Раздел 3. Наследование и интерфейсы

Контролируемые ИДК: ОПК-6.2

Тема 3.4. Пакеты и видимость членов класса

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Что делает метод equals (Object obj)?
 1. Сравнивает содержимое двух объектов
 2. Копирует одно значение в другое
 3. Удаляет объект
 4. Возвращает хэш-код объекта
2. Какой класс нельзя расширить?
 1. Final class
 2. Abstract class
 3. Any class can be extended
 4. Interface class
3. Каким образом реализуется множественное наследование в Java?
 1. Через использование интерфейсов
 2. Путём расширения нескольких классов
 3. Множественного наследования нет
 4. Никаким способом невозможно
4. Метод, который определяет поведение всех объектов класса, называют...
 1. Конструктор
 2. Свойство
 3. Метод
 4. Атрибут
5. Что возвращает метод toString() по умолчанию?
 1. Имя класса и адрес объекта в памяти
 2. Пустую строку
 3. Null
 4. Строку с именем объекта
6. Что делают ключевые слова public, private, protected?
 1. Контролируют память
 2. Управляют производительностью

3. Ограничивают доступность элементов класса
4. Устанавливают порядок наследования

Раздел 4. Коллекции и обобщения

Контролируемые ИДК: ОПК-6.2

Тема 4.3. Лямбда-выражения и функциональные интерфейсы

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Что обеспечивает связь разных классов через общий интерфейс?
 1. Инкапсуляция
 2. Композиция
 3. Полиморфизм
 4. Модульность
2. Какое свойство описывает способность изменять реализацию без изменения интерфейса?
 1. Адаптация
 2. Открытость-закрытость
 3. Замещение
 4. Подстановочность
3. Какая ошибка возникает при попытке присвоить целочисленную переменную вещественной без приведения типа?
 1. ArithmeticException
 2. ClassCastException
 3. NoError
 4. NumberFormatException
4. Что значит фраза "Java поддерживает одиночное наследование"?
 1. Класс может наследовать только одному родителю
 2. Нельзя реализовать два интерфейса
 3. Один класс может содержать только одну конструкцию
 4. Нет возможности наследоваться вообще
5. Что делает блок finally в try-catch?
 1. Выполняется перед началом блока catch
 2. Всегда выполняется после завершения блоков try/catch
 3. Никогда не выполняется
 4. Выполняется только при отсутствии исключений
6. Можно ли изменить длину массива в Java?
 1. Да, используя reallocate
 2. Нет, длина фиксирована
 3. Возможно изменение размера вручную
 4. Длина меняется автоматически

Раздел 5. Работа с файлами и ввод-вывод

Контролируемые ИДК: ОПК-6.1 ОПК-6.2

Тема 5.2. Форматы сериализации (XML, JSON)

Форма контроля/оценочное средство: Практическая работа

Вопросы/Задания:

1. Основы языка. Написание простых консольных приложений
2. Классы и объекты. Реализация классов и создание экземпляров
3. Конструкторы. Проектирование собственных конструкторов классов
4. Методы. Проекты с созданием методов

5. Наследование. Исследование иерархий классов
6. Абстрактные классы. Применение абстрактных классов и интерфейсов
7. Обработчик ошибок. Обработка исключительных ситуаций
8. Полиморфизм. Изучение различных форм полиморфизма
9. Коллекции. Работа с различными структурами данных
10. Потоки ввода-вывода. Программирование файлового ввода-вывода
11. Многопоточность. Запуск нескольких потоков одновременно
12. Сериализация. Сохранение состояния объектов в файл
13. Функциональные интерфейсы. Использование функциональных интерфейсов и лямбд
14. Сокращение данных с Stream API. Применение Stream API для анализа больших наборов данных
15. Простейшее сетевое приложение. Разработка простейших сервер-клиентских решений

Тема 5.3. Работа с сетью (Socket, HTTP)

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какой стандартный способ сравнения строк в Java?
 1. compareTo
 2. isEqual
 3. equalsIgnoreCase
 4. checkEquals
2. Что вернет следующий код: "Hello".equals("hello")?
 1. true
 2. false
 3. null
 4. undefined
3. Как назвать метод, который принимает аргумент своего же типа?
 1. Рекурсивный
 2. Самоссылающийся
 3. Перегруженный
 4. Рецессивный
4. Как проверить принадлежность объекта определенному типу?
 1. instanceof
 2. typeof
 3. typecheck
 4. instanceofof
5. Какой метод генерирует случайное число в диапазоне от 0 до 1?
 1. random.nextInt()
 2. Math.random()
 3. Random.nextDouble()
 4. System.getRandom()
6. Какая конструкция используется для ограничения изменения значения поля класса?
 1. public
 2. final
 3. protected
 4. private

8. Оценочные материалы промежуточной аттестации

Вопросы/Задания:

1. Что такое класс и объект?
2. Какие принципы относятся к основным принципам ООП?
3. Как объявляются переменные в Java?
4. Чем отличаются примитивные типы данных от типов-оберток?
5. Что такое конструкторы и зачем они нужны?
6. Опишите понятие инкапсуляции.
7. Для чего используется ключевое слово `final`?
8. Перечислите виды наследования в Java.
9. Когда применяется интерфейс?
10. Приведите примеры статических полей и методов.
11. Какой смысл имеет ключевое слово `abstract`?
12. Расскажите о понятии исключения (Exception) в Java.
13. Какие методы обработки исключений существуют?
14. Дайте определение полиморфизма.
15. Какова роль оператора `instanceof`?
16. Что такое переопределение метода?
17. Что такое коллекция в Java?
18. Зачем нужен интерфейс `Iterable`?
19. Какие стандартные структуры данных используются в Java?
20. Опишите механизм работы обобщённых коллекций (generics).
21. Как работает обработка потоков данных (I/O)?
22. Как осуществляется сериализация объектов?
23. Назначение класса `Scanner`.

24. В чём разница между методами write() и print()?
25. Опишите протокол TCP/IP.
26. Какие библиотеки применяются для сетевого взаимодействия?
27. Особенности анонимных классов.
28. Понятие лямбда-выражений и функционального интерфейса.
29. Использование Stream API для обработки коллекций.
30. Примеры многопоточного программирования в Java.

9. Материально-техническое и учебно-методическое обеспечение дисциплины

9.1. Перечень основной и дополнительной учебной литературы

Основная литература

1. Минакова О. В. Технологии программирования: паттерны проектирования в реализации JavaFX приложений: практикум: практикум / О. В. Минакова. - Технологии программирования: паттерны проектирования в реализации JavaFX приложений - Воронеж: Воронежский государственный технический университет, ЭБС АСВ, 2020. - 156 с. - 978-5-7731-0911-2. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/111505.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке
2. Саблина В. А. Основы программирования на JavaScript: учебное пособие: учебное пособие / В. А. Саблина, Е. А. Трушина. - Основы программирования на JavaScript - Рязань: Рязанский государственный радиотехнический университет, 2022. - 96 с. - 2227-8397. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/134868.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке
3. Курбатова И. В. Основы программирования на языке Java: учебное пособие для вузов: учебное пособие для вузов / И. В. Курбатова, А. В. Печуров. - Санкт-Петербург: Лань, 2024. - 348 с. - 978-5-507-48515-4. - Текст: электронный. // Издательство Лань: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/385928.jpg> (дата обращения: 16.03.2025). - Режим доступа: по подписке

Дополнительная литература

1. Блох Дж. Java. Эффективное программирование / Дж. Блох; перевод В. Стрельцов. - Java. Эффективное программирование - Саратов: Профобразование, 2024. - 310 с. - 978-5-4488-0127-3. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/145890.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке
2. Янцев В. В. JavaScript. Обработка событий на примерах / В. В. Янцев. - Санкт-Петербург: Лань, 2021. - 176 с. - 978-5-8114-7559-9. - Текст: электронный. // Издательство Лань: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/176881.jpg> (дата обращения: 16.03.2025). - Режим доступа: по подписке

9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся

Профессиональные базы данных

Не используются.

Ресурсы «Интернет»

1. <https://znanium.ru/> - Электронно-библиотечная система "Znanium"
2. <https://www.iprbookshop.ru/> - IPR BOOKS
3. <https://rusneb.ru/> - Национальная электронная библиотека (НЭБ)
4. <https://e.lanbook.com/> - Электронная библиотечная система "Лань"
5. <https://intuit.ru/> - Электронная образовательная среда "ИНТУИТ"

9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

Не используется.

Перечень информационно-справочных систем

(обновление выполняется еженедельно)

Не используется.

9.4. Оборудование и технические средства обучения

Учебная аудитория

Маркерная доска - 1 шт.

Моноблок - 96 шт.

Проектор - 7 шт.

Автоматизированное рабочее место (АРМ) - 15 шт.

Плоттер - 1 шт.

Сканер - 1 шт.

Смешанный тип

Автоматизированное рабочее место (АРМ) - 15 шт.

Доска настенная одноэлементная - 2 шт.

Проектор - 1 шт.

10. Методические указания по освоению дисциплины (модуля)

С целью успешного освоения данного предмета студенту необходимо выполнять домашние работы по указанным преподавателям темам и представленным на занятиях материалам. Для более успешного выполнения работы студенту предлагается использование учебно-методических материалов, соответствующих указанной теме.

Самостоятельная работа студентов описывается и регулируется методическими рекомендациями по дисциплине.

Описание возможностей изучения дисциплины лицами с ОВЗ и инвалидами

При необходимости обучения студентов-инвалидов и лиц с ограниченными возможностями здоровья аудиторные занятия могут быть заменены или дополнены изучением полнотекстовых лекций, презентаций, видео- и аудиоматериалов в электронной информационно-образовательной среде (ЭИОС) университета. Индивидуальные задания

подбираются в адаптированных к ограничениям здоровья формах (письменно или устно, в форме презентаций). Выбор методов обучения зависит от их доступности для инвалидов и лиц с ограниченными возможностями здоровья.

Рабочая программа дисциплины для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.