

Программа повышения квалификации «Программирование на Java. Уровень 1. Базовый курс»

состоит из 4 учебных модулей:

Модуль 1. Введение в Java технологии.

Модуль 2. Создание функций в Java, использование массивов, работа со строками и файлами.

Модуль 3. Основы объективно-ориентированного программирования.

Модуль 4. Дженерики и коллекции в Java.

Рабочая программа модуля 1. Введение в Java технологии.

Тема 1.1. Особенности языка программирования Java. Описание Java технологий. Использование интегрированной среды разработки (IDE).

Обзор языка программирования Java, описывая его историю, особенности и преимущества использования. Описание Java технологий. Обзор платформы Java: JVM, JRE, JDK. Различия между Java SE, Java EE и Java ME. Интегрированная среда разработки. Популярные среды разработки для Java: IntelliJ IDEA, Eclipse, NetBeans. Основные функции IDE. Установка ШК/IDE, настройка параметров среды.

Тема 1.2 Система типов Java. Операции с данными. Ссылочная модель в Java.

Примитивные типы данных. Преобразование простых типов. Обзор основных примитивных типов данных в Java и их особенностей. Явные и неявные преобразования. Операции присваивания и арифметические операции. Приоритет операций и примеры. Литералы и константы в Java. Практические примеры использования типов данных для решения задач. Ссылочная модель в Java и основы размещения/удаления объектов в памяти. Основные принципы работы ссылочной модели в Java. Создание и удаление объектов, размещение ссылки на объекты. Основы размещения и удаления ссылок в памяти. Сборщик мусора в Java, его принцип работы.

Тема 1.3 Ввод и вывод данных в Java. Сравнения и логические операции в Java.

Вывод данных через консоль. Форматированный вывод данных. Различные спецификаторы формата. Форматирование вывода данных в Java с учетом различных типов данных и требований к выводу. Объект Scanner в Java. Особенности ввода данных через консоль. Операторы сравнения (равно, не равно, больше, меньше, больше или равно, меньше или равно). Логические операторы (логические И, ИЛИ, НЕ). Оператор if для выполнения определенного блока кода. Оператор else. Написание программы с логической обработкой. Практические примеры использования условий для решения задач.

Тема 1.4. Операторы циклов for и while.

Цикл for. Цикл while. Рекомендации по использованию циклов. Операторы break и continue. Цикл do-while. Практические примеры использования циклов для решения задач.

Рабочая программа модуля 2. Создание функций в Java, использование массивов, работа со строками и файлами.

Тема 2.1. Определение функции в Java.

Создание собственной функции. Перегрузка функций: определение нескольких функций с одинаковым именем, но различающимися параметрами. Рекурсия: вызов функции из самой себя. Функциональные интерфейсы и лямбда-выражения. Возвращаемое значение функции: как функция может возвращать значения различных типов. Область видимости переменных в функциях.

Тема 2.2. Использование массивов.

Инициализация массивов: с помощью литералов, циклов или методов. Использование циклов для работы с массивами и коллекциями. Многомерные массивы. Проход по массиву: с помощью циклов for, while или for-each. Работа с строками и символами в массивах: преобразование строк в массивы символов и обратно.

Тема 2.3. Методы работы со строками в Java.

Введение в строки в Java. Создание и инициализация строк. Строковые литералы. Конструктор класса String. Длина строки. Сравнение строк. Разделение строки. Работа с классами StringBuilder и StringBuffer. Извлечение подстроки.

Тема 2.4. Обработка исключений и работа с файловой системой в Java. Обработка исключений при работе с массивами: `ArrayIndexOutOfBoundsException`, `NullPointerException`. Try-catch блоки. Множественный catch блок. Блок finally. Создание собственных исключений. Работа с файлами и директориями. Использование классов `File`, `FileInputStream`, `FileOutputStream`, `FileReader`, `FileWriter` для работы с файлами. Работа с байтовыми и символьными потоками. Работа с ресурсами: использование try-with-resources для автоматического закрытия ресурсов.

Рабочая программа модуля 3. Основы объектно-ориентированного программирования.

Тема 3.1. Создание собственного класса.

Определение класса: ключевое слово `class` с указанием имени класса. Поля класса: переменные, хранящие состояние объекта. Методы класса: функции, определяющие поведение объекта. Конструкторы. Модификаторы доступа: `public`, `private`, `protected` для управления доступом к членам класса. Наследование. Перегрузка методов и конструкторов. Применение класса.

Тема 3.2. Принципы объектно-ориентированного программирования.

Наследование. Инкапсуляция. Полиморфизм. Конструкторы и деструкторы. Перегрузка и переопределение методов. Пакеты и модули.

Тема 3.3. Понятия абстрактного класса и интерфейса.

Абстракция. Интерфейсы. Различия между абстрактным классом и интерфейсом. Абстрактные методы. Абстрактные классы. Интерфейсы: контракты, которые классы могут реализовывать, определяющие методы без реализации. Множественная реализация интерфейсов. Использование абстрактных классов и интерфейсов для достижения гибкости и расширяемости кода. Примеры использования абстрактных классов и интерфейсов в стандартной библиотеке Java.

Рабочая программа модуля 4. Дженерики и коллекции в Java.

Тема 4.1. Generics в коллекциях и методах.

Обобщенные классы и методы. Параметризация коллекций: использование обобщенных коллекций для хранения объектов определенного типа. Wildcard-символы: использование ? для работы с неопределенными типами в обобщенных коллекциях. Ограничения обобщений: использование extends и super для наложения ограничений на типы обобщений. Работа с обобщенными методами. Проверка типов во время компиляции. Преимущества использования обобщений.

Тема 4.2. Коллекции, List.

Различные типы коллекций: ArrayList, LinkedList, Vector, Stack и другие. Интерфейс List. Методы интерфейса List. Итерация по объектам List. Сортировка List. Преобразование List в массив и обратно. Использование List в качестве параметра методов: передача и возврат List из методов.

Тема 4.3. HashSet, HashMap и TreeMap.

HashSet. Реализация на основе HashMap. Методы. Внутреннее устройство. HashMap. Методы: put, get, remove, containsKey, containsValue. Разрешение коллизий. Производительность. TreeMap. Методы: put, get, remove, firstKey, lastKey. Реализация. Сложность операций: добавление, удаление, поиск в TreeMap имеют логарифмическую сложность.

Директор Высшей школе
управления государственными
и муниципальными финансами



В.В. Бондалетов