

## Java. Уровень 1. Основы программирования

Java – самый популярный объектно-ориентированный язык программирования. Java используется для создания серверного программного обеспечения, прикладных программ, разработки веб-сайтов, игр и т.д. Созданные на языке Java приложения могут успешно работать на любом компьютере независимо от его архитектуры.

**Дата проведения:** Открытая дата

**Вид обучения:** Курс повышения квалификации

**Формат обучения:** Онлайн-трансляция

**Срок обучения:** 5 дней

**Продолжительность обучения:** 40 часов

**Для участников предусмотрено:** Методический материал.

**Документ по окончании обучения:** По итогам обучения слушатели, успешно прошедшие итоговую аттестацию по программе обучения, получают Удостоверение о повышении квалификации в объеме 40 часов (в соответствии с лицензией на право ведения образовательной деятельности, выданной Департаментом образования и науки города Москвы).

### Для кого предназначен

Новичков или опытных программистов, желающих повысить свой уровень квалификации.

### Цель обучения

Научить программированию-Java-«с нуля». Выпускники-курса-смогут успешно продолжать освоение любой из высокоуровневых технологий семейства-Java: от приложений для мобильных устройств до Интернет-приложений и многопользовательских корпоративных систем.

### Особенности программы

Базовый синтаксис языка Java взят от-C/C++, а-значит, курс подойдет новичкам или опытным программистам, желающим повысить свой уровень квалификации.

Курс не-только дает фундаментальные основы, но-и-готовит к-последующей серьезной работе на-Java. Вы-подробно разберете все элементы и-конструкции языка Java и-возможности платформы JDK. Теория детально отрабатывается на-практике.

Обучение проводят высококвалифицированные преподаватели-практики, которые работают как с-Java-SE, так и-с-современными инструментальными средствами разработки ПО.

### Результат обучения

**В-результате обучения участники смогут:**

- разбираться в-Java технологиях;
- работать с-примитивными типами;
- использовать ветвление и-циклы;
- создавать методы на-Java;
- работать с-массивами и-со-строками;
- разрабатывать классы на-Java;
- использовать наследование и-полиморфизм;
- разрабатывать абстрактные классы и-интерфейсы;

- обрабатывать ошибки, возникающие в-программе;
- работать с-файловой системой;
- использовать новые возможности классов из-пакета NIO-2;
- пользоваться коллекциями Java;
- использовать в-программе параметризованные типы;
- создавать лямбда-выражения.

Это мероприятие можно заказать в корпоративном формате (обучение сотрудников одной компании).

# Программа обучения

## День-1.

### Введение в-Java технологии.

- Особенности языка программирования Java.
- Описание Java технологий.
- Использование интегрированной среды разработки.

### Практика:

- Установка JDK/IDE, настройка параметров среды.

### Введение в-язык программирования Java.

- Языковые лексемы Java.
- Введение в-систему типов языка Java.
- Работа с-примитивными типами и-константами.
- Операции языка Java.
- Преобразование простых типов.
- Контекстно-зависимое ключевое слово var.

### Методы и-операторы Java.

- Создание и-вызов методов.
- Перегрузка и-методы с-переменным числом аргументов.
- Операторы ветвления.
- Выражение switch.
- Операторы циклов.

## День-2.

### Создание и-использование массивов Java.

- Одномерные массивы.
- Многомерные массивы.
- Работа с-массивами и-класс Arrays.

**Практика:** Конвертор дня года в-месяц и-число.

### Работа со-строками в-Java.

- Основные методы класса String.
- Сложение и-преобразование строк.
- Классы динамических строк.
- Ввод данных с-клавиатуры и-класс Scanner.
- Регулярные выражения.

**Практика:** Конвертор целых чисел в-двоичную систему.

### Разработка классов на-Java.

- Обзор основных концепций ООП.
- Объявление класса.
- Члены класса и-создание объектов класса.

- Модификаторы доступа.
- Модификаторы final & static.
- Использование пакетов, директив импорта и-переменной среды CLASSPATH.

**Практика:** Разбор параметров командной строки.

### **День-3.**

#### **Наследование и-полиморфизм.**

- Наследование как механизм повторного использования кода.
- Конструктор при наследовании.
- Преобразование типов и-операция instanceof.
- Виртуальные методы и-позднее связывание.
- Абстрактные классы и-методы.

**Практика:**

- Разбор параметров командной строки. Часть-2. Наследование.

#### **Интерфейсы и-аннотации.**

- Концепция интерфейсов.
- Объявление интерфейса.
- Реализация интерфейса.
- Статические методы, методы по-умолчанию в-интерфейсах и-приватные методы.
- Использование и-создание аннотаций.

**Практика:**

- Сортировка массива объектов.

#### **Пакет java.lang.**

- Класс Object и-переопределение его методов.
- Метаданные и-рефлексия.
- Классы System и-Math.

### **День-4.**

#### **Обработка ошибок в-Java.**

- Концепция исключений в-Java.
- Использование операторов try, catch и-finally.
- Проверяемые и-непроверяемые исключения.
- Создание своих классов исключений.
- Оператор try для освобождения ресурсов.

#### **Потоки данных в-Java.**

- Обзор классов потоков.
- Работа с-байтовыми потоками.
- Работа с-потоками символов.
- Использование класса java.io.File.
- Сжатие файлов.
- Сериализация объектов в-Java.

**Практика:**

- Частотный анализ слов в-тексте.

#### **Работа с-файловой системой в-NIO 2.**

- Использование интерфейса Path.
- Работа с-атрибутами файлов.
- Основные возможности класса Files.
- Использование класса Files для обхода дерева каталогов.
- Мониторинг изменений в-файловой системе.

#### **Пакет java.util.**

- Форматирование данных.
- Работа с-датой и-временем.

- Класс Locale и-глобализация кода.
- Локализация и-класс ResourceBundle.
- Генерация псевдослучайных чисел.

#### **День-5.**

##### **Коллекции в-Java.**

- Иерархия классов коллекций.
- Концепция параметризованных типов данных.
- Работа с-параметризованным методов и-интерфейсом.
- Обзор возможностей списков, множеств и-словарей в-Java.

##### **Вложенные классы в-Java.**

- Внутренние классы.
- Вложенные классы.
- Анонимные классы.
- Перечисления в-Java.

##### **Лямбда-выражения.**

- Синтаксис лямбда-выражений.
- Ссылки на-методы.
- Функциональные интерфейсы.
- Новые возможности в-Java.

##### **Паттерны проектирования (дополнительный).**

- Обзор паттернов.
- Паттерн одиночка.
- Паттерн композиция.
- Паттерн наблюдатель.

**Практика:** Реализация паттерна наблюдатель.

# Преподаватели

## ШИШАКОВ Виталий Владимирович

Преподаватель-практик с 17-летним опытом работы в области информационных технологий.

#### **Образование:**

Аспирантура Физического факультета МГУ, кафедра Математического Моделирования и-Информатики.

#### **Опыт преподавания:**

Программирование (C++, Matlab, Python, Java, Web), теор.вер, и-другие специальные дисциплины.

**Опыт работы** в-различных отраслях-ИТ, связанных с-Web-программированием, системным администрированием, облачными инфраструктурами.