

студенты

15-55 лет



Разработка

программных проектов

с использованием JAVA

top

КОМПЬЮТЕРНАЯ
АКАДЕМИЯ



учёба с нуля
и бесплатно



диплом о проф.
переподготовке



трудоустройство
во время обучения



живая помощь
от преподавателей



платформа
с упражнениями



обновленная
программа 2025

736 академических
часов

1 год длительность
курса

368 часов аудиторных
занятий

368 часов самостоятельной
работы

Наши партнёры:

Yandex Cloud

hh карьерный
маркетплейс

Введение в язык программирования Java

- Сравнение Java с другими языками программирования.
- Алгоритм.
- Понятие блок-схемы.
- Программная среда Eclipse.
- JShell.

Переменные, типы данных, операторы

- Типы данных.
- Переменные.
- Константы и литералы.
- Операторы.
- Примеры построения программ с использованием блок-схем.

Логические операторы, операторы ветвлений, побитовые операторы

- Преобразование типов данных.
- Логические операторы.
- Таблица приоритетов операторов.
- Конструкции логического выбора. Операторы ветвлений.
- Побитовые операторы.

Циклы

- Циклы while, for, do...while.
- Работа с интегрированным отладчиком в Eclipse.

Строки, массивы одномерные, многомерные

- Работа со строками.
- Массивы.
- Алгоритмы суммирования, поиска и сортировки.
- Понятие сложности алгоритма.
- Многомерные массивы.

Методы (на примере статических методов)

- Методы.
- Область видимости.
- Рекурсия.

Объектно-ориентированное программирование

- Введение в объектно-ориентированное программирование.
- Понятие класса.
- Понятие объекта.
- Понятие члена класса, поля класса, метода класса.
- Спецификаторы доступа.
- Конструкторы объекта.
- Ключевое слово this.
- Перегрузка методов и конструкторов.
- Статические методы классов.
- Передача объектов в метод.
- Область видимости в методах классов.
- Наследование.
- Понятие интерфейса.
- Вложенные классы.
- Ключевое слово final.
- Сборка мусора.
- Пакеты.
- Шаблоны (generics).

Исключения

- Исключительная ситуация.
- Принципы обработки исключительных ситуаций.
- Понятие checked и unchecked.
- Ключевое слово try.
- Ключевое слово catch.
- Ключевое слово throw.
- Ключевое слово finally.
- Подробности использования исключительных ситуаций.
- Раскрутка стека вызовов.

Java Collection Framework

- Классы-обертки.
- Введение в JCF.
- Интерфейсы JCF.

- Создание коллекций с помощью фабричных методов.
- Классы JCF.

Аннотации, анонимные классы, Lambda-выражения

- Аннотации.
- Анонимные классы.
- Lambda-выражения.

Работа с файлами

- Знакомство с пакетом java.io.
- Потоки ввода/вывода.
- Сериализация объектов.

Stream API

- Stream API.
- Цели и задачи.
- Примеры использования.

Многопоточность

- Многопоточность в Java.
- Использование ExecutorService.
- Практические примеры.

Системы контроля версий

- Контроль версий.
- Необходимость контроля версий.
- Обзор систем контроля версий.
- Что такое Git?
- Цели и задачи Git.
- Основные термины.
- Использование внешних сервисов (github).

Работа в команде, управление программными проектами

- Что такое управление программными проектами?
- Причины возникновения дисциплины «Управление программными проектами».
- Диаграммы Ганта.
- Важные вопросы по управлению программными проектами.
- Анализ терминов предметной области.
- Характеристики проекта.
- Расходы, связанные с проектом.
- Общий обзор моделей и методологий процесса разработки.
- Подробнее о Scrum.

Использование JUnit

- Модульное тестирование.
- Цели и задачи модульного тестирования.
- Необходимость модульного тестирования.
- Обзор инструментов для модульного тестирования.
- Инструмент JUnit.

Паттерны проектирования

- Паттерны проектирования.
- Причины создания паттернов проектирования.
- Понятие паттерна проектирования.
- Принципы применения паттернов проектирования.
- Выбор паттернов проектирования.
- Принципы разделения паттернов на категории.
- Введение в UML.
- Использование UML при анализе паттернов проектирования.
- Порождающие паттерны.
- Структурные паттерны.
- Паттерны поведения.

Паттерны MVC

- Паттерн MVC.
- Цели и задачи паттерна Model-View-Controller.
- Model.
- View.
- Controller.
- Примеры использования паттерна MVC.

Принципы объектно ориентированного проектирования SOLID

- Обзор проблем, встречающихся при проектировании и разработке классов.

- Принципы проектирования классов SOLID.
- Примеры использования принципов SOLID.

Введение в теорию баз данных

- Введение в теорию баз данных.
- История СУБД Oracle.
- Архитектура СУБД Oracle.
- Версии СУБД Oracle.
- Утилиты.
- Демонстрация: Установка СУБД Oracle.
- Архитектура БД под управлением Oracle. Сравнение с другими СУБД.
- Демонстрация.
- Практическая работа: Создание базы данных с помощью файла конфигурации.
- Основы взаимодействия с Oracle.
- Запросы.
- Введение в язык структурированных запросов SQL.

Запросы SELECT, INSERT, UPDATE, DELETE

- Оператор SELECT.
- Ключевые слова IN, BETWEEN, LIKE.
- Оператор INSERT.
- Оператор UPDATE.
- Оператор DELETE.

Многотабличные базы данных

- Аномалии взаимодействия с однотабличной базой данных.
- Принципы создания многотабличной базы данных.
- Многотабличные запросы.

Функции агрегирования

- Функции агрегирования.
- Понятие группировки. Ключевое слово GROUP BY.
- Ключевое слово HAVING. Сравнительный анализ HAVING и WHERE.
- Подзапросы.

Объединения

- Операторы для использования в подзапросах.
- Объединение результатов запроса.
- Объединения JOIN.

Представления, хранимые процедуры, триггеры

- Понятие транзакции. Использование транзакций.
- Представления.
- Хранимые процедуры.
- Триггеры.

Введение в Web-технологии. Структура HTML. Форматирование текста с помощью HTML

- Введение в язык разметки. Язык разметки гипертекста HTML.
- Теги.
- Основные ошибки в записях тегов. Спецификации <!DOCTYPE HTML>.
- Валидация документа с помощью FireFox — дополнение HTML Validator.
- Понятие well-formed.
- Структура HTML5-документа.
- Кодировки страницы и теги <meta>.
- Классификация тегов: линейные и блочные.
- Модель форматирования текста: заголовки и абзацы. Элементы <p>, <h1>, <h2>. Выравнивание текста в блочных элементах: атрибут align.
- Классификация тегов: логическое и физическое форматирования.
- Практика: создание web-страницы.

Форматирование с помощью CSS. Списки. CSS отступы и поля

- CSS — каскадные таблицы стилей.
- Теги без форматирования: <div> — блочный, — линейный.
- Аналогия HTML и CSS на примере линейных и блочных тегов.
- Использование атрибутов class и id для задания стилей.

- Использование внешних CSS файлов стилей.
- Практика: форматирование текста с помощью CSS.
- Создание списков.
- Создание вложенных списков.
- Форматирование списков с помощью CSS.
- Списки определений: элементы <dl>, <dd>, <dt>.
- Управление отступами и полями.
- Практика: создание списков.

Графика в web-дизайне. Оптимизация графики. Гиперссылки. Принципы навигации web-сайта

- Форматы графических файлов в web.
- Тег и его атрибуты (src, alt, width, height, border).
- Фон страницы - свойство background.
- Общие сведения о гиперссылках.
- Абсолютная и относительная адресация.
- Создание меню с помощью структуры списков (,), его форматирование. Свойство display. Преобразование ссылки в блочный элемент.
- Псевдоклассы.
- CSS свойство cursor.
- Практика: работа по разработке галереи изображений.
- Свойства CSS3.
- Работа с мультимедиа.

Таблицы

- Создание простейшей таблицы. Теги <table>, <tr> и <td>.
- Объединение ячеек: атрибуты colspan, rowspan.
- Теги логического структурирования таблиц: <thead>, <tbody>, <tfoot>. Теги логического группирования столбцов: <colgroup>, <col>.
- Управление рамками таблицы: атрибуты frame, rules.
- Практика: создание сложных таблиц.
- Основы табличной верстки. Пример табличной верстки: ее минусы.

Позиционирование. Верстка web-страниц блоками

- Свойство position.
- Свойства visibility, overflow.
- Основы верстки блоками. Правила верстки.
- Рассмотрение простейших структур страниц.
- Резиновая структура. Блоки с отрицательными margin.

Формы. Фреймы

- Управляющие элементы форм.
- Создание форм с помощью HTML.
- Элементы форм из HTML5.
- Валидация форм с помощью HTML5.
- Форматирование элементов форм с помощью CSS.
- Фреймы и их структура (теоретические сведения).

Введение в JavaScript

- Сценарии, выполняемые на стороне клиента.
- JavaScript.
- История создания JavaScript.
- Различия между JavaScript и Java, JScript, ECMAScript.
- Версии JavaScript.
- Document Object Model.
- Browser Object Model.
- Внедрение в HTML документы. Редакторы кода JavaScript.
- Тег <noscript>.
- Основы синтаксиса.
- Переменные. Правила именования переменных.
- Типы данных.
- Операторы.
- Ввод/вывод данных. Диалоговые окна.
- Условия.
- Циклы.
- Функции.
- Объект Arguments.
- Область видимости переменной. Ключевое this.
- Рекурсия.

Объект. Массивы. Объект Array. Строки. Объект String. Объект Date. Объект Math. Введение в объектно-ориентированное программирование

- Объекты.
- Массивы.
- Строки.
- Задержки и интервалы. Периодический вызов функций.
- Объект Date. Обработка даты и времени.
- Объект Math. Свойства и методы. Случайные числа.
- ООП.
- Три фундаментальных принципа ООП.
- Понятие класса и объекта в терминах JavaScript.
- Свойства.
- Методы.
- Свойства-аксессуары.
- Конструктор.
- Понятие «prototype».
- Наследование.

Обработка событий

- События.
- Обработчик события.
- Обработка событий в сценариях.
- Управление стилями элементов web-страницы.
- Объект Event и его свойства.
- Обработчики событий по умолчанию (стандартные обработчики), запрет вызова стандартного обработчика.
- Объект Image. Управление рисунками и ролловерами.

Browser Object Model. Document Object Model

- Browser Object Model.
- Объекты Browser Object Model.
- Document Object Model.
- Отличия DOM и BOM.
- Представление HTML-документа в виде дерева.
- Объекты модели DOM. Иерархия узлов.
- Свойства и методы модели DOM. Модель событий DOM.
- Изменение дерева DOM.
- Знакомство с объектами Document и Link.
- Управление выделением и текстовым диапазоном: объекты Selection и TextRange.
- Особенности DOM в HTML5.

Формы

- Применение форм. Размещение элементов формы в HTML.
- Коллекция Forms. Создание и программирование элементов формы.

Проверка достоверности форм. Использование Cookie

- Объект RegExp. Правила записи регулярных выражений.
- Методы объектов String и RegExp для работы с регулярными выражениями.
- Проверка достоверности данных формы.
- Cookie.
- Преимущества и недостатки cookie.
- Создание, использование и удаление cookie.

Рисование с помощью Canvas, поддержка медиа-возможностей

- Canvas.
- Базовые возможности Canvas.
- Поддержка медиа возможностей.

JSON, AJAX

- Цели и задачи JSON.
- Синтаксис JSON.
- Объект JSON.
- Настройка пользовательской сериализации в JSON. Метод toJSON.
- Синхронные и асинхронные запросы.
- AJAX.
- Объект XMLHttpRequest.
- Методы и свойства XMLHttpRequest.

- Понятие HTTP-заголовка.
- Использование метода GET. URL-кодирование.
- Использование метода POST.

Введение в сетевые технологии

- Сетевое и серверное программирование.
- Цели и задачи сетевого и серверного программирования.
- Типы сетей.
- Модель OSI.
- Базовые термины.
- Схема взаимодействия клиента и сервера.

Сетевое взаимодействие

- Обзор пакета java.net.
- Класс InetAddress.
- Класс Socket.
- Класс ServerSocket.
- Класс DatagramSocket.
- Класс DatagramPacket.
- Практическая работа. Создание файлового сервера.

Введение в разработку серверных решений с использованием Java

- Введение в серверное программирование.
- Краткий обзор полезных утилит и библиотек.
- Понятие сервлета.

Взаимодействие с источниками данных

- Источники данных.
- JDBC.
- Работа с JDBC.

JavaServer Pages, Tags в JSP

- JSP.
- Цели и задачи JSP.
- История возникновения JSP.
- Понятие директивы.
- Обработка ошибок в JSP.
- JSP и Model View Controller.
- Expression Language в JSP.
- JavaBean.
- Java Standard Tag Library.
- Различные виды Tags.
- Использование Conditional Tags.
- Использование Iteration Tags.
- Примеры использования других Tags.
- Custom Tags.
- Tag Files.
- JSP Fragment.
- Примеры использования.

Введение в Spring

- Цели и задачи Spring.
- История возникновения.
- Архитектура Spring.
- REST и SOAP.
- Spring MVC.
- Архитектура Spring MVC.
- Примеры использования.
- Spring Boot.
- Spring Security.
- Spring Data.
- Микросервисная архитектура.

Введение в Hibernate, Spring Data

- Что такое Hibernate?
- Цели и задачи Hibernate.
- История возникновения.
- Архитектура Hibernate.
- Примеры использования.
- Что такое Spring Data?
- Цели и задачи Spring Data.
- Основы принципы использования Spring Data.
- Примеры использования Spring Data.

ИТОГОВАЯ ПРАКТИЧЕСКАЯ РАБОТА

ПО ОКОНЧАНИИ КУРСА СЛУШАТЕЛЬ БУДЕТ:

- Понимать и уметь использовать фундаментальные принципы создания программ с использованием Java.
- Уметь создавать, компилировать, и производить отладку приложений в IDE Eclipse.
- Уметь проектировать и реализовывать различные алгоритмы.
- Разбираться и использовать в проектах принципы ООП.
- Уметь проектировать классы различной степени сложности.
- Использовать механизмы generics для построения шаблонных классов.
- Понимать механизмы многопоточности Java.
- Уметь пользоваться системой контроля версий.
- Использовать принципы командного взаимодействия.
- Применять паттерны проектирования.
- Использовать юнит-тестирование.
- Разбираться в языке структурированных запросов SQL.
- Уметь создавать многотабличные запросы.
- Понимать принципы работы подзапросов и функций агрегирования.
- Разбираться в тонкостях реализации клиентских сценариев под разные браузеры.
- Понимать особенности применения HTML5 по отношению к JavaScript.
- Уметь сериализовать и парсить данные, используя JSON.
- Владеть принципами создания асинхронных запросов с помощью AJAX.
- Понимать фундаментальные принципы создания серверных решений с использованием Java.
- Уметь создавать, компилировать, и отлаживать веб-приложения.
- Уметь взаимодействовать с источниками данных.
- Использовать сетевые механизмы.
- Понимать и применять паттерн MVC.
- Уметь создавать JSP-решения.
- Использовать фреймворк Spring.
- Использовать библиотеку Hibernate.



17

лет на рынке



4,8

рейтинг
на Я. Картах



380

филиалов в России



2 800

преподавателей



120 000+

студентов



24 000+

отзывов



Также доступно
онлайн-обучение

*данные действительны
на январь 2025 года