ШАГ компьютерная АКАДЕМИЯ

Программа курса
«Язык программирования Java»

Продолжительность курса: 184 пары

Расписание: 2 пары аудиторных занятий два раза в неделю и минимум 6 часов на выполнение домашних заданий

В рамках курса вы научитесь:

- создавать ПО с использованием Java;
- уметь проектировать классы различной степени сложности и создавать иерархии классов для решения практических задач;
- использовать фундаментальные принципы создания серверных решении с использованием Java;
- понимать механизмы многопоточности Java.

Преимущества курса Java

- Программа обучения построена в соответствии с требованиями IT-рынка.
- Курс нацелен на получение практических навыков разработки, в программе предусмотрено большое количество практических работ и финальный проект, которые станут вашим портфолио.
- Курс читают преподаватели-практики.
- Срок обучения в 1 год позволяет получить профессиональные практические навыки и умения в области создания программных проектов.
- Обучение проходит в малокомплектных группах.
- Занятия проводятся на современном оборудовании в комфортных аудиториях.
- Тренинги по развитию soft skills.
- Курс по техническому английскому языку.

Выпускники курса смогут работать в IT-компаниях и в стартап-проектах.

Требования к поступающим:

- возраст от 15 до 55 лет;
- уровень подготовки: уверенное владение ПК.

Тематический план

Язык программирования Java

Версия 2.0.1
Продолжительность курса – 80 пар

Цель курса
Обучить слушателя языку программирования Java. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

По окончании курса слушатель будет:

- понимать фундаментальные принципы создания программ с использованием Java;
- уметь создавать, компилировать, и отлаживать проекты в IDE Eclipse;
- уметь проектировать и реализовывать различные алгоритмы;
- использовать механизмы условий и циклов;
- применять массивы для хранения данных;
- уметь использовать алгоритмы сортировки и поиска данных;
- разбираться в принципах ООП;
- уметь проектировать классы различной степени сложности;
- создавать иерархии классов для решения практических задач;
- использовать механизмы Generics для построения шаблонных классов;
- уметь порождать и обрабатывать исключительные ситуации;
- выбирать и использовать классы JCF;
- сохранять и читать информацию из файлов;
- понимать механизмы многопоточности Java:
- использовать лямбды;
- уметь пользоваться системой контроля версий;
- понимать основы командного взаимодействия
- применять паттерны проектирования;
- использовать юнит-тестирование.

По окончании данного курса студент сдает практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену должны быть сданы все домашние и практические задания. Практическое задание должно охватывать максимум материала из различных разделов курса.

Тематический план

Модуль 1. Введение в язык программирования Java

1. Вступление.
   - История и этапы развития языка Java.
   - Сравнительный анализ языка Java с другими языками программирования.
   - Что такое виртуальная машина?
   - Что такое байт-код?
2. Алгоритм.
   - Понятие алгоритма.
   - Примеры использования алгоритмов в реальной жизни.
   - Типы алгоритмов. Линейный, разветвленный, циклический.
3. Понятие блок-схемы.
   - Базовые обозначения в блок-схемах.
   - Блок начала алгоритма.
   - Блок завершения алгоритма.
   - Блок ввода данных.
   - Блок вывода данных.
   - Блок вычислений.
   - Простейшие примеры использования блок-схем.
4. Программная среда Eclipse.
   - Инсталляция.
   - Основы работы с IDE Eclipse.
   - Создание проекта.
   - Добавление файла к проекту.
   - Обзор альтернативных средств разработки.
   - Запуск простейшего приложения.
5. JShell.
   - Что такое JShell?
   - Цели и задачи JShell
   - Примеры использования JShell.

Модуль 2. Переменные, типы данных, операторы

1. Типы данных.
   - Понятие типа данных. Размер, диапазон значений.
   - Целые типы данных.
   - Типы данных для работы с дробными числами.
   - Символьный тип данных.
   - Логический тип данных.
   - Перечислимый тип данных.
2. Переменная.
   - Необходимость использования переменных.
   - Идентификаторы.
   - Ключевые слова.
   - Синтаксис объявления переменных.
3. Константы и литералы.
   - Необходимость применения.
   - Синтаксис объявления.
4. Операторы.
   - Понятие оператор.
   - Типы операторов:
     - арифметические операторы;
     - логические операторы;
     - операторы ветвлений;
     - унарные операторы;
     - бинарные операторы;
     - тернарный оператор.
   - Оператор присваивания.
   - Арифметические операторы:
     - оператор сложения;
     - оператор вычитания;
     - оператор умножения;
     - оператор деления;
     - оператор деления по модулю;
     - инкремент. Постфиксная и префиксная форма;

- декремент. Постфиксная и префиксная форма;
- сокращенные формы.

5. Примеры построения программ с использованием блок-схем.

Модуль 3. Логические операторы, операторы ветвлений, побитовые операторы

1. Преобразование типов данных.
   - Необходимость использования.
   - Неявное преобразование типов.
   - Явное преобразование типов.
2. Логические операторы.
   - Знакомство с логическими операциями.
   - Таблица результатов применения логических операций.
   - «Логическое отрицание». Оператор !
   - «Логическое И». Оператор &&
   - «Логическое ИЛИ». Оператор ||
3. Таблица приоритетов операторов.
4. Конструкции логического выбора. Операторы ветвлений.
   - Оператор ветвления if.
   - Оператор ветвления if – else.
   - Лестница if – else if.
   - Обозначение условий в блок-схемах. Блок условия.
   - Обозначение объединения ветвей в блок-схемах.
   - Примеры построения программ с использованием операторов ветвлений на языке блок-схем.
   - Понятие составного оператора.
   - Тернарный оператор.
   - Оператор множественного выбора – switch.
5. Побитовые операторы.
   - Системы исчисления двоичная, восьмеричная, шестнадцатеричная.
   - Цели и задачи битовых операций.
   - Битовое «И».
   - Битовое «ИЛИ».

- Битовое «ИСКЛЮЧАЮЩЕЕ ИЛИ».
- Битовое отрицание.
- Битовые сдвиги.

Модуль 4. Циклы

1. Циклы.
   - Необходимость использования циклов. Примеры использования.
   - Цикл while.
   - Цикл for.
   - Цикл do-while.
   - Обозначение циклов в блок-схемах. Блок цикла.
   - Операторы break и continue.
   - Примеры построения программ с использованием циклов на языке блок-схем.
   - Вложенные циклы. Примеры использования.
2. Работа с интегрированным отладчиком в Eclipse.
   - Что такое отладчик. Цели и задачи отладчика.
   - Запуск программы по шагам.
   - Окна для работы с отладчиком. Окна переменных, локальных переменных, памяти.
   - Исполнение одного шага.
   - Установка точки останова (breakpoint).

Модуль 5. Строки, массивы одномерные, многомерные

1. Работа со строками.
2. Массивы.
   - Что такое массивы?
   - Необходимость использования массивов.
   - Синтаксис объявления одномерного массива.
   - Схема размещения массивов в памяти.
   - Индексация элементов массива.
   - Примеры использования массивов на языке блок-схем.

3. Алгоритмы суммирования.
4. Алгоритмы поиска.
   - Линейный.
   - Бинарный.
5. Алгоритмы сортировки.
   - Пузырьковая сортировка.
   - Сортировка выбором.
   - Сортировка вставками.
   - Другие алгоритмы сортировки (быстрая сортировка и т. д.).
6. Понятие сложности алгоритма.
7. Многомерные массивы.
   - Многомерные массивы. Цели и задачи их использования.
   - Двумерные массивы как частный случай многомерных.
   - Синтаксис объявления многомерного массива.
   - Примеры использования многомерных массивов.

Модуль 6. Методы (на примере статических методов)

1. Методы.
   - Что такое метод?
   - Необходимость использования методов.
   - Синтаксис объявления методов.
   - Использование ключевого слова void при работе с методами.
   - Вызов метода.
   - Аргументы.
   - Возврат значения из метода (return).
2. Область видимости.
   - Понятие области видимости.
   - Примеры использования областей видимости.
3. Рекурсия.

Модуль 7. Объектно-ориентированное программирование

1. Введение в объектно-ориентированное программирование.
   - Инкапсуляция.
   - Полиморфизм.
   - Наследование.
2. Понятие класса.
3. Понятие объекта.
4. Понятие члена класса, поля класса, метода класса.
5. Спецификаторы доступа.
6. Конструкторы объекта.
   - Что такое конструктор?
   - Цели и задачи конструктора.
   - Примеры создания конструкторов.
7. Ключевое слово this.
8. Перегрузка методов и конструкторов.
9. Статические методы классов.
   - Что такое статический метод класса?
   - Отличие статического и обычного метода класса.
   - Примеры использования статических методов.
10. Передача объектов в метод.
11. Область видимости в методах классов.
12. Наследование.
    - Спецификаторы доступа при наследовании.
    - Ключевое слово super.
    - Порядок вызова конструкторов.
    - Переопределение методов.
    - Динамическая диспетчеризация методов.
    - Абстрактный класс.
13. Понятие интерфейса.
    - Что такое интерфейс?
    - Реализация интерфейса.
    - Использование реализации интерфейса через ссылки.

- Вложенные интерфейсы.
- Переменные и интерфейсы.

14. Вложенные классы.
15. Ключевое слово final.
    - Использование final для классов.
    - Использование final для методов.
16. Сборка мусора.
    - Что такое сборка мусора?
    - Принцип работы сборщика мусора.
    - Что такое финализатор?
    - Метод finalize.
    - Принципы создания финализатора.
17. Пакеты.
18. Шаблоны (Generics).
    - Что такое шаблоны?
    - Цели и задачи шаблонов.
    - Шаблонные классы.
    - Шаблонные методы.
    - Шаблонные конструкторы.
    - Шаблонные интерфейсы.
    - Шаблоны и наследование.

Модуль 8. Исключения

1. Что такое исключительная ситуация?
2. Принципы обработки исключительных ситуаций.
3. Понятие checked и unchecked исключений.
    - Что такое checked и unchecked исключения?
    - Отличия и принципы использования.
4. Ключевое слово try.
5. Ключевое слово catch.
6. Ключевое слово throw.
7. Ключевое слово finally.

8. Подробности использования исключительных ситуаций.
9. Раскрутка стека вызовов.

Модуль 9. Динамические структуры данных. JavaCollectionFramework

1. Понятие динамической структуры данных.
2. Стек.
   - Понятие стека.
   - Принцип LIFO.
   - Пример создания и практического использования стека.
3. Очереди.
   - Понятие очереди.
   - Типы очередей:
     - обычная очередь. Принцип FIFO;
     - кольцевая очередь;
     - очередь с приоритетами;
     - примеры создания и использования очередей.
4. Другие динамические структуры данных.
   - Списки.
   - Бинарные деревья.
   - Другие.
5. Введение в JCF.
   - Причины создания.
   - Обзор.
6. Интерфейсы JCF.
   - Collection
   - Comparator
   - Enumeration
   - EventListener
   - Iterator
   - List
   - ListIterator
   - Map
   - Map.Entry

- Observer
- RandomAccess
- Set
- SortedMap
- SortedSet

7. Создание коллекций с помощью фабричных методов.
8. Классы JCF.
   - AbstractCollection
   - AbstractList
   - AbstractMap
   - AbstractSequentialList
   - AbstractSet
   - ArrayList
   - Arrays
   - BitSet
   - Collections
   - Dictionary
   - HashMap
   - HashSet
   - Hashtable
   - IdentityHashMap
   - LinkedHashMap
   - LinkedHashSet
   - LinkedList
   - Stack
   - TreeMap
   - TreeSet
   - Vector

Модуль 10. Аннотации, анонимные классы, Lambda выражения

1. Аннотации.
2. Анонимные классы.

3. Lambda выражения.
   - Что такое лямбда-выражения?
   - Цели и задачи лямбда-выражений.
   - Синтаксис лямбда-выражений.
   - Примеры создания лямбда-выражений.

Модуль 11. Работа с файлами

1. Знакомство с пакетом java.io.
2. Потоки ввода/вывода.
   - Потоки ввода/вывода.
   - Фильтрованные потоки.
   - Канальные потоки.
   - Буферизированные потоки.
   - Файловые потоки.
   - Потоки для работы с файлами.
   - Потоки, размещаемые в оперативной памяти.
3. Сериализация объектов.
   - Понятие сериализации.
   - Граф сериализации.
   - Использование сериализации.

Модуль 12. Stream API

1. Stream API.
2. Что такое Stream API?
3. Цели и задачи.
4. Примеры использования.

Модуль 13. Многопоточность

1. Многопоточность в Java.
   - Что такое многопоточность?
   - Класс Thread.
   - Интерфейс Runnable.
   - Приоритеты потоков.

- Синхронизация потоков:
  - проблемы, возникающие при синхронизации потоков;
  - метод wait;
  - метод notify;
  - метод notifyall.

2. Использование ExecutorService.
3. Практические примеры.

Модуль 14. Системы контроля версий

1. Что такое контроль версий?
2. Зачем нужен контроль версий.
3. Обзор систем контроля версий:
   - CVS;
   - SVN;
   - Git;
   - Другие системы контроля версий.
4. Git.
   - Что такое Git?
   - Цели и задачи Git?
   - Основные термины:
     - репозиторий;
     - коммит;
     - ветка;
     - рабочий каталог.
5. Операции с Git.
   - Установка.
   - Создание репозитория.
   - Добавление файла в репозиторий.
   - Запись коммита в репозиторий.
   - Получение текущего состояния рабочего каталога.
   - Отображение веток.
   - Операции с накопительным буфером.
   - git remote.

- git push.
- git pull.
- Другие операции.

6. Использование внешних сервисов (github).

Модуль 15. Работа в команде, управление программными проектами

1. Что такое управление программными проектами?
2. Причины возникновения дисциплины управление программными проектами.
3. Диаграммы Ганта.
4. Важные вопросы по управлению программными проектами.
   - Что такое проект и программный проект?
   - Что такое жизненный цикл процесса разработки программного обеспечения?
   - Что такое управление проектами?
   - Что такое одиночная разработка?
   - Что такое командная разработка?
   - Анализ проблем одиночной и командной разработки программного обеспечения.
5. Анализ терминов предметной области.
   - Процесс.
   - Проект.
   - Персонал.
   - Продукт.
   - Качество.
6. Характеристики проекта.
   - Тип проекта.
   - Цель проекта.
   - Требования к качеству.
   - Требования к бюджету.
   - Требования по срокам завершения.

7. Расходы, связанные с проектом.
   - Прямые.
   - Непрямые.
8. Общий обзор моделей и методологий процесса разработки.
   - Фазы процесса:
     - определение требований;
     - проектирование;
     - конструирование («реализация», «кодирование»);
     - интеграция;
     - тестирование и отладка («верификация»);
     - инсталляция;
     - поддержка.
   - Водопадная модель.
   - Спиральная модель.
   - Итеративная модель:
     - Agile;
     - Scrum;
     - XP.
   - RUP.
   - MSF.
   - Анализ существующих моделей и методов.
9. Подробнее о Scrum.
   - Что такое Scrum?
   - Причины возникновения Scrum.
   - Роли в Scrum:
     - владелец продукта;
     - команда;
     - scrum мастер.
   - Бэклог продукта:
     - что такое бэклог продукта?
     - как создавать бэклог?
     - как оценивать задачи в бэклоге?
     - что такое scrum-доска?
     - примеры создания бэклога.

- Спринт:
  - что такое спринт?
  - планирование спринтов;
  - ежедневный скрам;
  - обзор спринта;
  - ретроспективное собрание.

Практическое задание: Необходимо провести симуляцию работы команды по методологии Scrum. Например, это может быть так называемое скрам-лего.
Подробно тут: Scrum Simulation with LEGO Bricks.

Модуль 16. Использование jUnit

1. Что такое модульное тестирование?
2. Цели и задачи модульного тестирования.
3. Необходимость модульного тестирования.
4. Обзор инструментов для модульного тестирования.
5. Инструмент jUnit.
   - Что такое jUnit?
   - История создания jUnit.
   - Практические примеры использования jUnit.

Модуль 17. Паттерны проектирования

1. Что такое паттерны проектирования.
2. Причины возникновения паттернов проектирования.
3. Понятие паттерна проектирования.
4. Принципы применения паттернов проектирования.
5. Принципы выбора паттернов проектирования.
6. Принципы разделения паттернов на категории.
7. Введение в UML.
   - Диаграмма классов.
   - Диаграмма объектов.
   - Диаграмма взаимодействия.
8. Использование UML при анализе паттернов проектирования.
   - Диаграмма классов.

- Диаграмма объектов.
- Диаграмма взаимодействия.

9. Порождающие паттерны.
   - Что такое порождающий паттерн?
   - Цели и задачи порождающих паттернов.
   - Обзор порождающих паттернов.
   - Разбор порождающих паттернов:
     - Abstract Factory:
       - цель паттерна;
       - причины возникновения паттерна;
       - структура паттерна;
       - результаты использования паттерна;
       - практический пример использования паттерна.
     - Builder:
       - цель паттерна;
       - причины возникновения паттерна;
       - структура паттерна;
       - результаты использования паттерна;
       - практический пример использования паттерна.
     - Factory Method:
       - цель паттерна;
       - причины возникновения паттерна;
       - структура паттерна;
       - результаты использования паттерна;
       - практический пример использования паттерна.
     - Prototype:
       - цель паттерна;
       - причины возникновения паттерна;
       - структура паттерна;
       - результаты использования паттерна;
       - практический пример использования паттерна.
     - Singleton:
       - цель паттерна;
       - причины возникновения паттерна;

- структура паттерна;
- результаты использования паттерна;
- практический пример использования паттерна.

10. Структурные паттерны.
    - Что такое структурный паттерн?
    - Цели и задачи структурных паттернов.
    - Обзор структурных паттернов.
    - Разбор структурных паттернов:
        - Adapter;
        - Composite;
        - Facade;
        - Proxy;
        - другие структурные паттерны.
11. Паттерны поведения:
    - Что такое паттерны поведения?
    - Цели и задачи паттернов поведения.
    - Обзор паттернов поведения.
    - Разбор паттернов поведения:
        - Command
        - Iterator
        - Observer
        - Strategy
        - другие структурные паттерны.

Модуль 18. Паттерн MVC

1. Что такое паттерн MVC?
2. Цели и задачи паттерна Model-View-Controller.
3. Model:
    - Что такое Model?
    - Цели и задачи Model.
4. View:
    - Что такое View?
    - Цели и задачи View.

5. Controller:
   - Что такое Controller?
   - Цели и задачи Controller.
6. Примеры использования паттерна MVC.

Модуль 19. Принципы проектирования классов SOLID

1. Обзор проблем, встречающихся при проектировании и разработке классов.
2. Принципы проектирования классов SOLID:
   - принцип единственности ответственности (The Single Responsibility Principle);
   - принцип открытости/закрытости (The Open Closed Principle);
   - принцип подстановки Барбары Лисков (The Liskov Substitution Principle);
   - принцип разделения интерфейса (The Interface Segregation Principle);
   - принцип инверсии зависимостей (The Dependency Inversion Principle).
3. Примеры использования принципов SOLID.

Модуль 20. Экзамен

Теория баз данных

Версия 1.0.0
Продолжительность курса – 16 пар

Цель курса
Ввести слушателя в системы управления базами данных. Обучить языку структурированных запросов SQL; принципам нормализации; работе с хранимыми процедурами, триггерами, видами, пользовательскими функциями. Получить теоретические и практические знания о СУБД.

По окончании курса слушатель будет:
- разбираться в языке структурированных запросов SQL;
- уметь создавать многотабличные запросы;
- понимать принципы работы подзапросов и функций агрегирования;
- уметь производить нормализацию баз данных;
- использовать хранимые процедуры, триггеры, виды.

По окончании данного курса студент сдает практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену должны быть сданы все домашние и практические задания.
Практическое задание должно охватывать максимум материала из различных разделов курса.

Тематический план

Модуль 1. Введение в теорию баз данных

1. Введение в теорию баз данных.
   - История и этапы развития.
   - Понятия база данных и система управления базами данных.
   - Сравнение существующих моделей баз данных:
     - файловая модель;
     - сетевая модель;
     - иерархическая модель;
     - реляционная модель;
     - объектно-ориентированная модель.
   - Понятие реляционной модели баз данных.
   - Двенадцать правил Кодда.
2. История СУБД Oracle.
3. Архитектура СУБД Oracle.
4. Версии СУБД Oracle.
5. Утилиты.
   - SQL Plus.
   - Database Configuration Assistant.
   - Administration Assistant for Windows.
   - Net Configuration Assistant.
6. Демонстрация: Инсталляция СУБД Oracle.
7. Архитектура БД под управлением Oracle. Сравнение с другими СУБД.
   - Создание базы данных с помощью Database Configuration Assistant.
   - Создание базы данных с помощью файла конфигурации.
8. Демонстрация.
   - Создание базы данных и управления базами данных с помощью Database Configuration Assistant.
   - Создание базы данных с помощью файла конфигурации.

Практическая работа: создание базы данных с помощью файла конфигурации.

9. Основы взаимодействия с Oracle.
   - Объекты Oracle.

- Таблицы:
  - первичный ключ;
  - значение по умолчанию;
  - уникальность.
- Типы данных:
  - целочисленные типы;
  - типы данных для хранения текста;
  - вещественные типы данных;
  - типы для хранения даты и времени;
  - типы данных с фиксированной точкой;
  - другие типы данных.
- Понятие индекса:
  - цели и задачи индексов;
  - внутреннее устройство индексов.

10. Запросы.
11. Введение в язык структурированных запросов SQL.
    - Язык SQL. Стандарты языка SQL.
    - Понятия DDL, DML, DCL.

Модуль 2. Запросы SELECT, INSERT, UPDATE, DELETE

1. Оператор SELECT.
   - Предложение SELECT.
   - Предложение FROM.
   - Предложение WHERE.
   - Предложение ORDER BY.
2. Ключевые слова IN, BETWEEN, LIKE.
3. Оператор INSERT.
4. Оператор UPDATE.
5. Оператор DELETE.

Модуль 3. Многотабличные базы данных

1. Аномалии взаимодействия с однотабличной базой данных.
   - Аномалии обновления.

- Аномалии вставки.
- Аномалии обновления.
- Аномалии удаления.

2. Принципы создания многотабличной базы данных.
   - Причины создания многотабличной базы данных.
   - Внешний ключ.
   - Связи. Типы связей.
   - Целостность данных.
   - Нормализация:
     - необходимость нормализации;
     - понятие нормальной формы;
     - первая нормальная форма;
     - вторая нормальная форма;
     - третья нормальная форма;
     - нормальная форма Бойса-Кодда.
3. Многотабличные запросы.
   - Принципы создания многотабличного запроса.
   - Декартовое произведение.

Модуль 4. Функции агрегирования

1. Функции агрегирования.
   - Функция COUNT.
   - Функция AVG.
   - Функция SUM.
   - Функция MIN.
   - Функция MAX.
2. Понятие группировки. Ключевое слово GROUP BY.
3. Ключевое слово HAVING. Сравнительный анализ HAVING и WHERE.
4. Подзапросы.
   - Необходимость создания и использования подзапросов.
   - Сравнение подзапросов и многотабличных запросов.
   - Принцип работы подзапросов.

Модуль 5. Объединения

1. Операторы для использования в подзапросах.
   - Оператор EXISTS.
   - Операторы ANY/SOME.
   - Оператор ALL.
2. Объединение результатов запроса.
   - Принципы объединения.
   - Ключевое слово UNION.
   - Ключевое слово UNION ALL.
3. Объединения JOIN.
   - Понятие inner join.
   - Понятие left join.
   - Понятие right join.
   - Понятие full join.

Модуль 6. Представления, хранимые процедуры, триггеры

1. Понятие транзакции. Использование транзакций.
2. Представления.
   - Создание представлений.
   - Модификация представлений.
   - Удаление представлений.
   - Изменения данных через представления.
3. Хранимые процедуры.
4. Триггеры.

Модуль 7. Экзамен

Разработка веб-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3

Версия 3.0.0
Продолжительность курса – 14 пар

Цель курса
Обучить слушателя созданию и верстке статических web-страниц с использованием технологий HTML5, CSS3. Сложить для слушателя целостное представление о технологической цепочке создания web-сайтов и сформировать понимание актуальных тенденций развития web-технологий. Научить слушателя выбирать наиболее подходящий способ для создания web-страниц. Научить тестировать и проверять код web-страниц.

По окончании курса слушатель будет:

- знать и уметь применять основы HTML – теги, атрибуты и способы структурирования содержимого web-страниц для создания форматированных документов;
- знать и уметь применять основы CSS – значения, списки, цвета, шрифты и другие метрики форматирования;
- владеть навыками проверки и отладки кода web-документов;
- владеть навыками формирования содержимого web-документов для различных экранов – от стандартных браузеров до мобильных устройств;
- владеть навыками быстрого и качественного форматирования сложных web-документов;
- знать основы HTML5 и CSS3.

По окончании данного курса студент сдает все практические задания курса. На основании всех сданных заданий выставляется оценка по предмету.

Обязательное практическое задание в конце курса, выполняемое студентом дома: создание web-сайта с последующим размещением в Internet. Основные требования: блочная верстка, валидный код.

Тематический план

Модуль 1. Введение в Web-технологии. Структура HTML. Форматирование текста при помощи HTML . . . . . **2 пары**

Модуль 2. Форматирование при помощи CSS. Списки. CSS отступы и поля . . . . . **3 пары**

Модуль 3. Графика в web-дизайне. Оптимизация графики. Гиперссылки. Принципы навигации web-сайта . . . . . **3 пары**

Модуль 4. Таблицы . . . . . **2 пары**

Модуль 5. Позиционирование. Верстка web-страниц блоками . . . . . **2 пары**

Модуль 6. Формы. Фреймы . . . . . **2 пары**

Модуль 1. Введение в Web-технологии. Структура HTML. Форматирование текста при помощи HTML

1. Введение в предмет.
2. Введение в языки разметки. Язык разметки гипертекста HTML.
   - Развитие HTML, версии. Версия HTML5.
   - Вопросы межбраузерной совместимости. Война браузеров.
   - W3C.
3. Теги – основной элемент структуры HTML. Правила записи тегов и их атрибутов в стандарте HTML5 на примере тегов <b>, <i>, <u>, <font>, <sup>, <sub>,
. Синтаксические отличия HTML4, XHTML, HTML5.
4. Основные ошибки в записях тегов.
   - Спецификации <!DOCTYPE HTML>.
   - Валидация документа при помощи FireFox – дополнение HTML Validator.
   - Понятие well-formed.
   - Прародители HTML5: SGML и XML.
5. Структура HTML5 документа.
   - Основные элементы и их назначения.
   - Новые теги задания структуры: <header>, <nav>, <section>, <article>, <aside>, <footer>. Доступность новых тегов в современных браузерах. Отображение новых тегов в устаревших браузерах.
6. Кодировки страницы и теги <meta>.
   - Применение тега <meta> – задание информации о странице (expires, refresh, autor, copyright, keywords, description).
   - Задание кодировки страницы при помощи тега <meta>.
   - Символьные подстановки и кодировки.
7. Классификация тегов: линейные и блочные.
   - Линейные: <b>, <i>, <u>, <font>.
   - Блочные: <p>, <h1>..<h6>.
8. Модель форматирования текста: заголовки и абзацы. Элементы <p>, <h1>..<h6>. Выравнивание текста в блочных элементах: атрибут align.
9. Классификация тегов: логическое и физическое форматирования.
   - Теги физического форматирования: <b>, <i>.
   - Теги логического форматирования: , . Их отличие.

- Краткий обзор основных тегов логического форматирования: <abbr>, <acronym>, <cite>, <code>, <del>, <dfn>, <ins>.

Практика: создание простейшей web-страницы.

Модуль 2. Форматирование при помощи CSS. Списки. CSS отступы и поля

1. CSS – каскадные таблицы стилей.
   - Введение. Обзор версий. Назначение: HTML служит для задания структуры, CSS – для форматирования.
   - Встраивание CSS в HTML при помощи атрибута style. Правила записи CSS свойств.
2. Теги без форматирования – блочный, – линейный.
3. Аналогия HTML и CSS на примере линейных и блочных тегов.
   - Тег <font> – свойства color, font-size, font-family.
   - Тег <b> – свойства font-weight.
   - Тег <i> – свойства font-style.
   - Тег <u> – свойства text-decoration.
   - Тег <sup>, <sub> – свойства vertical-align.
   - Атрибут align – свойства text-align.
   - Сокращенная запись свойства font.
   - Дополнительные свойства CSS для форматирования текста: letter-spacing, line-height, text-intend, text-transform, white-space, word-spacing.
4. Использование атрибутов class и id для задания стилей.
   - Создание стилей для тегов, классов, идентификаторов внутри тега <style>. Понятие селекторов. Правило записи селекторов: селектор тегов, селектор классов, селектор идентификаторов, универсальный селектор *.
   - Приоритет использования стилей (tag / class / id / style). Повышение приоритета правилом !important.
   - Наследуемость стилей. Стандартные значения свойств.
   - Отслеживание стилей при помощи средства разработки firebug (дополнение для Firefox).
5. Использование внешних CSS файлов стилей.
   - Подключение CSS файлов при помощи тега <link> и инструкции @import.
   - CSS файлы и кэш браузера.

Практика: форматирование текста при помощи CSS.

6. Создание списков.
   - Неупорядоченные списки: элементы <ul>, <li>.
   - Упорядоченные списки: элементы <ol>, <li>.
   - Атрибуты type, value, start.
7. Создание вложенных списков.
8. Форматирование списков при помощи CSS.
   - Свойства list-style-type, list-style-image, list-style-position.
   - Сокращенная запись свойства list-style.
   - Оформление многоуровневых списков. Вложенные селекторы.
9. Списки определений: элементы <dl>, <dd>, <dt>.
10. Управление отступами и полями.
    - Свойство margin и его потомки margin-left, margin-top, margin-right, margin-bottom.
    - Свойство padding и его потомки padding-left, padding-top, padding-right, padding-bottom.
    - Отличие padding от margin и их назначения.
    - Отмена отступов по умолчанию у некоторых тегов: <body>, <h1>..<h6>, <p>.

Практика: создание списков.

Модуль 3. Графика в web-дизайне. Оптимизация графики. Гиперссылки. Принципы навигации web-сайта

1. Форматы графических файлов в Web.
2. Тег <img /> и его атрибуты (src, alt, width, height, border).
   - Свойство border – аналог атрибута border.
   - Задание свойств margin, padding, border для изображения.
   - Выравнивание изображений на странице при помощи атрибута align. Аналог атрибута align – свойство float.
3. Фон страницы – свойство background.
   - Задание фона в виде цвета: background-color. Обязательное задание фона для элемента <body>.
   - Задание фона в виде изображения: background-image, background-repeat, background-position, background-attachment.

- Изображения и кэш браузера.

4. Общие сведения о гиперссылках.
   - Тег <a> и его атрибуты (href, target).
   - Эргономика, удобство навигации.
5. Абсолютная и относительная адресация.
   - Организация внешних ссылок.
   - Организация внутренних ссылок с помощью элемента <a>. Атрибуты id и name.
   - Организация «смешанного» перехода (на указанный элемент во внешнем HTML-документе).
   - Графические ссылки. Отмена границ у ссылок.
6. Создание меню при помощи структуры списков (<ul>, <li>), его форматирование. Свойство display. Преобразование ссылки в блочный элемент.
7. Псевдоклассы.
   - Псевдоклассы ссылок: active, hover, link, visited.
   - Псевдоклассы для обычных элементов: first-child, first-line, first-letter.
8. CSS свойство cursor.

Практика: работа по разработке галереи изображений.

9. Свойства из CSS3.
   - Работа с фоном: создание градиентов, изменение размеров фона – свойства background и background-size.
   - Работа с границами: скругленные края у блоков – свойства border-radius.
   - Задание полупрозрачности элементам страниц – свойство opacity.
   - Полная поддержка селекторов CSS 2.1.
10. Работа с мультимедиа.
    - Вставка видео на странице посредством тега <video>.
    - Вставка аудио на странице посредством тега <audio>.
    - Создание изображений и анимации посредством тега <canvas>.
    - Использование SVG формата.

Модуль 4. Таблицы

1. Создание простейшей таблицы. Теги <table>, <tr> и <td>.
   - Атрибуты border, cellspacing cellpadding. Их возможные аналоги CSS: border, padding.

- Указание ширины и высоты ячейки: атрибуты width, height. Правила задания ширины и высоты. Аналоги CSS: свойства width, height.
- Выравнивание данных в таблице: атрибуты align и valign. Аналоги CSS: свойства text-align, vertical-align.
- Управление цветом фона и цветом рамок таблицы (отдельной строки, отдельной ячейки).
- Использование изображений в качестве фона таблицы (отдельной строки, отдельной ячейки).

2. Объединение ячеек: атрибуты colspan, rowspan.
3. Теги логического структурирования таблиц: <thead>, <tbody>, <tfoot>. Теги логического группирования столбцов: <colgroup>, <col>.
4. Управление рамками таблицы: атрибуты frame, rules.

Практика: создание сложных таблиц.

5. Основы табличной верстки. Пример табличной верстки: ее минусы и плюсы.

Модуль 5. Позиционирование. Верстка web-страниц блоками

1. Свойство position.
   - Рассмотрение позиционирования: relative и absolute.
   - Свойства top, left, bottom, right.
2. Свойства visibility, overflow.

Практика.

3. Основы верстки блоками. Правила верстки.
   - Вложение блоков.
   - Задание ширины и высоты блокам при помощи свойства width и height.
   - Обтекание блоков. Отмена обтекания блоков. Свойства float и clear.
   - Правила задания отступов и полей.
   - Задание минимальной высоты и ширины блока: свойства min-height, min-width. Задание этих свойств в браузере IE6.
   - Выравнивание внутри блоков (margin, text-align, line-height, position). Кроссбраузерность выравниваний.
4. Рассмотрение простейших структур страниц и элементов.
   - Структура сайта фиксированного размера.
5. Резиновая структура. Блоки с отрицательными margin.

Модуль 6. Формы. Фреймы

1. Введение в формы.
2. Управляющие элементы форм.
   - Кнопки (отправки, сброса, пр.).
   - Флажки.
   - Кнопки с зависимой фиксацией (радиокнопки).
   - Всплывающие списки.
   - Текстовый ввод.
   - Выбор файлов.
   - Скрытые управляющие элементы.
3. Создание форм при помощи HTML.
   - Элемент <form>.
   - Элемент <input>.
   - Элемент <button>.
   - Элементы <select>, <optgroup> и <option>.
   - Элемент <textarea>.
   - Метки <label>.
   - Структура форм: <fieldset> и <legend>.
4. Элементы форм из HTML5.
5. Валидация форм при помощи HTML5.
6. Форматирование элементов форм при помощи CSS.
7. Фреймы и их структура (теоретические сведения).
   - Тег <iframe>.
   - Использование Спецификации <!DOCTYPE HTML> для фреймов.
   - Вредность использования фреймов.
   - Применение тега <iframe> в визуальных редакторах WYSIWYG.

Разработка клиентских сценариев с использованием JavaScript

Версия 2.0.0
Продолжительность курса – 16 пар

Цель курса
Обучить студента разработке клиентских сценариев с исполь-зованием JavaScript. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

По окончании курса слушатель будет:

- владеть базовыми конструкциями языка JavaScript, такими как переменные, условия, циклы, строки, массивы функции, и т.д.;
- знаком с ООП и его основными понятиями;
- уметь обрабатывать возникающие ошибки;
- разбираться в понятиях событие, обработчик события;
- создавать функции-обработчики различных событий;
- понимать отличия BOM и DOM;
- уметь взаимодействовать с объектами из BOM и DOM;
- разбираться в тонкостях реализации клиентских сценариев под разные браузеры;
- владеть принципами создания форм и анализа данных пользователя с использованием регулярных выражений;
- уметь сохранять пользовательские данные с помощью механизма cookie;
- понимать особенности применения HTML5 по отношению к JavaScript;
- уметь сериализовать и парсить данные используя JSON;
- владеть принципами создания асинхронных запросов при помощи Ajax.

По окончании данного курса студент сдает все практические задания курса. На основании всех сданных заданий выставляется оценка по предмету.

Тематический план

Модуль 1. Введение в JavaScript. . . . . **2 пары**

Модуль 2. Объект. Массивы. Объект Array. Строки. Объект String. Объект Date. Объект Math. Введение в объектно-ориентированное программирование . . . . . **2 пары**

Модуль 3. Обработка событий . . . . . **2 пары**

Модуль 4. Browser Object Model. Document Object Model. . . . . **2 пары**

Модуль 5. Формы . . . . . **2 пары**

Модуль 6. Проверка достоверности форм. Использование Cookie . . . . . **2 пары**

Модуль 7. Рисование с помощью canvas, поддержка медиа-возможностей . . . . . **2 пары**

Модуль 8. JSON, Ajax . . . . . **2 пары**

Модуль 1. Введение в JavaScript

1. Сценарии, выполняемые на стороне клиента.
2. Что такое JavaScript?
3. История создания JavaScript.
4. Различия между JavaScript и Java, JScript, ECMAScript.
5. Версии JavaScript.
6. Понятие Document Object Model.
7. Понятие Browser Object Model.
8. Внедрение в HTML документы. Редакторы кода JavaScript.
9. Тег <noscript>.
10. Основы синтаксиса.
    - Регистрозависимость.
    - Комментарии.
    - Ключевые и зарезервированные слова.
11. Переменные. Правила именования переменных.
12. Типы данных.
13. Операторы.
    - Арифметические операторы.
    - Операторы отношений.
    - Логические операторы.
    - Оператор присваивания.
    - Битовые операторы.
    - Приоритет операторов.
    - Оператор typeof.
14. Ввод/вывод данных. Диалоговые окна.
15. Условия.
    - Что такое условие?
    - if
    - if else
    - Тернарный оператор ?:
    - switch
16. Циклы.

- Что такое цикл?
- while
- do while
- for
- break
- continue
- Понятие метки.

17. Что такое функция?
    - Синтаксис объявления функции.
    - Параметры функции.
    - Возвращаемое значение функции. Ключевое слово return.
18. Объект arguments.
    - Цель и задачи объекта.
    - Свойство length.
19. Область видимости переменной. Ключевое this.
20. Рекурсия.

Модуль 2. Объект. Массивы. Объект Array. Строки. Объект String. Объект Date. Объект Math. Введение в объектно-ориентированное программирование

1. Объекты.
    - Что такое объект?
    - Введение в объектный тип данных.
    - Объект Object.
    - Ключевое слово new.
    - Понятие свойства.
    - Добавление свойств. Синтаксис добавления свойств.
    - Синтаксис обращения к свойствам.
2. Массивы.
    - Что такое массив?
    - Объект Array.
    - Создание массива.

- Обращение к элементам массива.
- Свойства и методы Array.

3. Строки.
   - Объект String.
   - Свойства и методы String.
4. Задержки и интервалы. Периодический вызов функций.
5. Объект Date. Обработка даты и времени.
6. Объект Math. Свойства и методы. Случайные числа.
7. Что такое ООП?
8. Три фундаментальных принципа ООП.
   - Инкапсуляция.
   - Наследование.
   - Полиморфизм.
9. Понятие класса и объекта в терминах JavaScript.
10. Свойства.
11. Методы.
12. Свойства-акссесоры.
    - get–свойства (геттеры).
    - set–свойства (сеттеры).
13. Конструктор.
14. Понятие prototype.
    - Что такое prototype.
    - Цели и задачи prototype.
15. Наследование.

Модуль 3. Обработка событий

1. Что такое событие?
2. Что такое обработчик события?
3. Обработка событий в сценариях.
4. Управление стилями элементов web-страницы.
5. Объект event и его свойства.

6. Обработчики событий по умолчанию (стандартные обработчики), запрет вызова стандартного обработчика.
7. Объект Image. Управление рисунками и ролловерами.

Модуль 4. Browser Object Model. Document Object Model

1. Что такое Browser Object Model?
2. Объекты Browser Object Model.
   - Объект Window. Открытие, перемещение и изменение размера окон.
   - Объект Navigator. Управление браузером.
   - Объект Screen. Свойства экрана.
   - Объекты Location и History. Перемещение по страницам.
   - Коллекция Frames. Управление фреймами.
3. Что такое Document Object Model?
4. Отличия DOM от BOM.
5. Представление HTML-документа в виде дерева.
6. Объекты модели DOM. Иерархия узлов.
7. Свойства и методы модели DOM. Модель событий DOM.
8. Изменение дерева DOM.
9. Знакомство с объектами Document и Link.
10. Управление выделением и текстовым диапазоном: объекты Selection и TextRange.
11. Особенности DOM в HTML5.

Модуль 5. Формы

1. Применение форм. Размещение элементов формы в HTML.
2. Коллекция Forms. Создание и программирование элементов формы.
   - Кнопки: элементы Button, Submit, Reset.
   - Текстовые поля: элементы Text, Password, File Upload, Textarea.
   - Скрытое поле формы: общее понятие об элементе Hidden.
   - Флажок: элемент Checkbox.
   - Переключатель: элемент Radio.
   - Список: элементы Select, Option.

Модуль 6. Проверка достоверности форм. Использование Cookie

1. Объект RegExp. Правила записи регулярных выражений.
2. Методы объектов String и RegExp для работы с регулярными выражениями.
3. Проверка достоверности данных формы.
4. Что такое cookie?
5. Преимущества и недостатки cookie.
6. Создание, использование и удаление cookie.

Модуль 7. Рисование с помощью canvas, поддержка медиа-возможностей

1. Что такое canvas?
2. Базовые возможности.
   - Заливка.
   - Операции с графическими примитивами. Рисование точек, линий, прямоугольников, кругов, кривых Безье и т. д.
   - Вывод текста.
   - Вывод изображений.
   - Работа с тенями и градиентом.
3. Поддержка медиа возможностей.
   - Использование тега <video>.
   - Использование тега <audio>.
   - Практические примеры.

Модуль 8. JSON, Ajax

1. Что такое JSON?
2. Цели и задачи JSON.
3. Синтаксис JSON.
   - Переменные.
   - Объекты.
   - Массивы.
4. Объект JSON.
   - Что такое сериализация?

- Что такое парсинг?
- Методы stringify и parse.

5. Настройка пользовательской сериализации в JSON. Метод toJSON.
6. Синхронные и асинхронные запросы.
7. Что такое Ajax?
8. Объект XMLHttpRequest.
   - Создание через ActiveX объект.
   - Создание через объект XMLHttpRequest.
9. Методы и свойства XMLHttpRequest.
10. Понятие HTTP заголовка.
11. Использование метода GET. URL кодирование.
12. Использование метода POST.

Разработка серверных решений с использованием Java

Версия 2.0.0
Продолжительность курса – 58 пар

Цель курса
Обучить слушателя созданию серверных решений с использованием Java. Научить выбирать правильные механизмы и конструкции для решения той или иной практической задачи.

По окончании курса слушатель будет:

- понимать фундаментальные принципы создания серверных решений с использованием Java;
- уметь создавать, компилировать, и отлаживать веб-приложения;
- уметь взаимодействовать с источниками данных;
- использовать сетевые механизмы;
- уметь создавать сервлеты;
- понимать и применять паттерн MVC;
- уметь создавать JSP решения;
- применять механизмы cookies и сессий;
- использовать фреймворк Spring;
- использовать библиотеку Hibernate.

По окончании данного курса студент сдает практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену должны быть сданы все домашние и практические задания. Практическое задание должно охватывать максимум материала из различных разделов курса.

Тематический план

Модуль 1. Введение в сетевые технологии . . . . . **2 пары**

Модуль 2. Сетевое взаимодействие . . . . . **4 пары**

Модуль 3. Введение в разработку серверных решений с использованием Java . . . . . **6 пар**

Модуль 4. Взаимодействие с источниками данных . . . . . **6 пар**

Модуль 5. JavaServer Pages, Tags в JSP . . . . . **8 пар**

Модуль 6. Введение в Spring . . . . . **20 пар**

Модуль 7. Введение в Hibernate, Spring Data. . . . . **10 пар**

Модуль 8. Экзамен . . . . . **2 пары**

Модуль 1. Введение в сетевые технологии

1. Что такое сетевое и серверное программирование?
2. Цели и задачи сетевого и серверного программирования.
3. Что такое сеть?
4. Типы сетей.
5. Модель OSI.
6. Базовые термины.
   - Сетевые протоколы.
   - IP адрес.
   - Сокет.
   - Порт.
   - Веб-сервер.
7. Схема взаимодействия клиента и сервера.
   - Понятие http request/response.
   - Методы HTTP:
     - GET;
     - POST;
     - HEAD;
     - PUT;
     - DELETE;
     - Другие.

Модуль 2. Сетевое взаимодействие

1. Обзор пакета java.net.
2. Класс InetAddress.
3. Класс Socket.
4. Класс ServerSocket.
5. Класс DatagramSocket.
6. Класс DatagramPacket.

Практическая работа. Создание файлового сервера.

Модуль 3. Введение в разработку серверных решений с использованием Java

1. Введение в серверное программирование.
   - Что такое серверное решение?
   - Что такое веб-приложение?
   - Чем отличается клиентская и серверная часть приложения?
   - Какне механизмы предоставляет Java для создания веб-приложений?
   - Какие утилиты полезны для создания веб-приложений на Java?
2. Краткий обзор полезных утилит и библиотек.
   - Что такое Maven?
   - Что такое TomCat?
   - Что такое JBoss?
   - Что такое Spring?
   - Что такое Hibernate?
3. Понятие сервлета.
   - Что такое сервлет?
   - Цели и задачи сервлета.
   - Каркас сервлета.
   - Базовые интерфейсы сервлета.
   - Базовые классы сервлета.
   - Пример создания простого сервлета.
   - Настройка сервлета.
   - Взаимодействие сервлета и клиента (http request / response).
   - Примеры создания сложных сервлетов.

Модуль 4. Взаимодействие с источниками данных

1. Источники данных.
   - Что такое источник данных?
   - Какие бывают источники данных?
   - База данных как источник данных.
2. JDBC.
   - Что такое JDBC?

- История возникновения JDBC.
- Версии JDBC.
- Использование JDBC для доступа к различным СУБД.

3. Работа с JDBC.
   - Соединение с СУБД.
   - Получение данных из базы данных.
   - Сохранение данных в базу данных.
   - Обновление данных в базе данных.
   - Примеры использования JDBC в сервлетах.

Модуль 5. JavaServer Pages, Tags в JSP

1. Что такое JSP?
2. Цели и задачи JSP.
3. История возникновения JSP.
4. Понятие директивы.
5. Обработка ошибок в JSP.
6. JSP и Model View Controller.
7. Expression Language в JSP.
   - Что такое Expression Language?
   - Цели и задачи Expression Language.
   - Примеры использования.
8. JavaBean.
   - Что такое JavaBean.
   - Цели и задачи JavaBean.
   - Примеры использования.
9. Java Standard Tag Library.
   - Что такое Java Standard Tag Library?
   - Цели и задачи Java Standard Tag Library.
   - Понятие Tag.
10. Различные виды Tags.
    - Core Tags.
    - Formatting Tags.

- SQL Tags.
- XML Tags.
- JSTL functions.

11. Использование Conditional Tags.
12. Использование Iteration Tags.
13. Примеры использования других Tags.
14. Что такое Custom Tags?
15. Что такое Tag Files?
16. Что JSP Fragment?
17. Примеры использования.

Модуль 6. Введение в Spring

1. Что такое Spring?
2. Цели и задачи Spring.
3. История возникновения.
4. Архитектура Spring.
5. Spring MVC.
6. REST и SOAP.
   - Что такое REST?
   - Что такое SOAP?
   - Практические примеры.
7. Spring MVC.
8. Архитектура Spring MVC.
8. Примеры использования.
9. Spring Boot.
   - Что такое Spring Boot?
   - Цели и задачи Spring Boot.
   - Примеры использования Spring Boot.
10. Spring Security.
    - Что такое Spring Security?
    - Цели и задачи Spring Security.
    - Примеры использования Spring Security.

11. Spring Data.
    - Что такое Spring Data?
    - Цели и задачи Spring Data.
    - Примеры использования Spring Data.
12. Микросервисная архитектура.
    - Что такое микросервис?
    - Идеология микросервисной архитектуры.
    - Spring и микросервисы.
    - RabbitMQ и микросервисы.
    - Примеры создания микросервисов.

Модуль 7. Введение в Hibernate, Spring Data

1. Что такое Hibernate?
2. Цели и задачи Hibernate.
3. История возникновения.
4. Архитектура Hibernate.
5. Примеры использования.

Модуль 8. Экзамен