ДЕТАЛЬНАЯ ПРОГРАММА ОБУЧЕНИЯ

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ШАГ компьютерная АКАДЕМИЯ

ШАГ компьютерная АКАДЕМИЯ

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

СОДЕРЖАНИЕ

ШАГ компьютерная АКАДЕМИЯ

Программа курса «Основы программирования на языке C++»

Основи програмування на мові C++
Fundamentals of C++ Programming

Цель курса

Обучить слушателя основам программирования на языке C++. Научить студента мыслить алгоритмически. Научить использовать блок-схемы, условия, циклы, массивы, функции, указатели и другие базовые конструкции языка программирования C++.
Объектно-ориентированное программирование не является целью данного курса.

По окончании курса слушатель будет:

- понимать, что такое алгоритм;
- уметь разрабатывать алгоритмы;
- уметь строить блок-схемы;
- использовать циклы, условия и другие базовые конструкции;
- оперировать основами языка программирования C++;
- понимать и использовать отладчик;
- создавать функции;
- использовать одномерные и многомерные массивы;
- применять указатели и разбираться в тонкостях арифметики указателей;
- уметь работать с динамически выделенной памятью;
- использовать алгоритмы поиска и сортировки данных;
- оперировать структурами, объединениями, битовыми полями;
- сохранять и загружать данные из файловой системы;
- применять битовые операции для воздействия на отдельные биты переменных.

По окончании данного курса студент сдает экзамен (по материалам курса), содержащий теоретическую и практическую часть. Для допуска к экзамену, должны быть сданы все домашние и практические задания.

Перед началом данного предмета необходимо предоставить студентам доступ к следующим курсам Cisco Networking Academy:

- Programming Essentials in C;
- Programming Essentials in C++.

Тематический план

Модуль 1. Введение в язык программирования C++

Модуль 2. Переменные и типы данных

Модуль 3. Логические операторы и операторы ветвлений

Модуль 4. Циклы, использование отладчика

Модуль 5. Массивы: одномерные и многомерные

Модуль 6. Функции

Модуль 7. Указатели

Модуль 8. Строки

Модуль 9. Многомерные динамические массивы

Модуль 10. Структуры

Модуль 11. Препроцессор в приложениях

Модуль 12. Использование файловой системы

Модуль 13. Экзамен

Модуль 1

Введение в язык программирования C++

1. Введение.
   - История и этапы развития языка C++.
   - Сравнительный анализ языка C++ с другими языками программирования.
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 Microsoft Visual C++.
   - Инсталляция.
   - Основы работы с IDE VC++.
   - Создание проекта.
   - Добавление файла к проекту.
   - Обзор альтернативных средств разработки.
5. Первая программа.
   - Построение первой программы на языке блок-схем.
   - Анализ первой программы.
6. Классификация символов языка.
7. Лексемы.
8. Понятие библиотеки.
9. Анализ понятий: компилятор, линковщик, интерпретатор.

10. Организация вывода данных в консоль.
11. Понятие ESCAPE-последовательности.
    - ESCAPE последовательность \n.
    - ESCAPE последовательность \t.
    - ESCAPE последовательность \b.
    - ESCAPE последовательность \».
    - ESCAPE последовательность \\.
    - ESCAPE последовательность \a.
12. “Raw” строки.
    - Что такое “raw” строки.
    - Примеры использования “raw” строк.
13. Комментарии.
    - Однострочные.
    - Многострочные.

Модуль 2

Переменные и типы данных

1. Типы данных.
    - Понятие типа данных. Размер, диапазон значений.
    - Целые типы данных. Типы данных long long, unsigned long long.
    - Типы данных для работы с дробными числами.
    - Символьный тип данных. Типы char16_t, char32_t.
    - Логический тип данных.
    - Перечислимый тип данных (enum).
2. Переменная.
    - Необходимость использования переменных.
    - Идентификаторы.
    - Ключевые слова.
    - Синтаксис объявления переменных.
    - Инициализация переменной. Списковая инициализация переменной.
3. Организация ввода данных с консоли.
4. Константы и литералы.
    - Необходимость применения.
    - Синтаксис объявления.

5. Операторы.
   - Понятие оператора.
   - Типы операторов.
   - Арифметические операторы.
   - Логические операторы.
   - Операторы ветвлений.
   - Унарные операторы.
   - Бинарные операторы.
   - Тернарный оператор.
   - Оператор присваивания.
   - Арифметические операторы.
   - Оператор сложения.
   - Оператор вычитания.
   - Оператор умножения.
   - Инкремент. Постфиксная и префиксная форма.
   - Декремент. Постфиксная и префиксная форма.
   - Сокращенные формы.
6. Примеры построения программ с использованием блок-схем.

Модуль 3

Логические операторы и операторы ветвлений

1. Преобразование типов данных.
   - Необходимость использования.
   - Неявное преобразование типов.
   - Списковая инициализация, сужение и 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. Понятие enum.
   - Понятие enum как перечислимого типа.
   - Синтаксис объявления enum.
   - Использование enum для switch-конструкций.

Модуль 4

Циклы, использование отладчика

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 Microsoft Visual C++.
   - 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).
- Установка умной точки останова (smart breakpoint).

Модуль 5

Массивы: одномерные и многомерные

1. Массивы.
   - Что такое массивы. Необходимость их использования.
   - Синтаксис объявления одномерного массива.
   - Способы инициализации массива.
   - Схема размещения массивов в памяти.
   - Индексация элементов массива.
   - Примеры использования массивов на языке блок-схем.
2. Алгоритмы суммирования.
3. Алгоритмы поиска (линейный, бинарный).
4. Алгоритмы сортировки.
   - Пузырьковая сортировка.
   - Сортировка выбором.
   - Сортировка вставками.
5. Многомерные массивы.
   - Многомерные массивы. Цели и задачи их использования.
   - Двумерные массивы, как частный случай многомерных.
   - Синтаксис объявления многомерного массива.
   - Примеры использования многомерных массивов.
6. Понятие статического выделения памяти.

Модуль 6

Функции

1. Функции.
   - Необходимость использования функций.
   - Синтаксис объявления функции.
   - Использование ключевого слова void при работе с функциями.
   - Вызов функции.
   - Аргументы функции.

- Возврат значения из функции (оператор return).
- Хвостовой возвращаемый тип.
- Понятие области видимости. Локальные и глобальные переменные. Классы памяти.
- Передача массива в функцию.
- Прототип функции.
- Аргументы по умолчанию.
- Встраивание (inline функции).
- Перегрузка функций.
- Шаблоны функций:
  - понятие шаблона. Определение и объявление шаблона. Инстанционирование шаблона;
  - синтаксис объявления шаблонной функции;
  - ключевые слова class и typename;
  - отличия обычной и шаблонной функции;
  - примеры создания шаблонных функций (например, Максимум, Минимум, Сортировка, Поиск и так далее);
  - перегрузка шаблонных функций;
  - ключевые слова auto и decltype. Автоматическое выведение типа;
  - использование decltype в шаблонных функциях;
  - хвостовой возвращаемый тип функций (использование ->) и decltype.
- Рекурсия:
  - что такое рекурсия;
  - цели и задачи рекурсии;
  - примеры рекурсивных функций;
  - алгоритм быстрой сортировки.
- Функция, принимающая неограниченное количество элементов.

Модуль 7

Указатели

1. Указатели.
   - Необходимость использования указателей.
   - Адрес переменной. Оператор &.
   - Синтаксис объявления указателя.
   - Косвенная адресация или оператор разыменования.
   - Принцип работы оператора присвоения для указателей.

- Анализ использования NULL и nullptr.
- Связь массивов и указателей.
- Операции над указателями:
  - арифметические операции;
  - логические операции.
- Примеры работы с указателями.
- Константный указатель и указатель на константу.
- Понятие стека и динамической памяти.
- Средства языка для работы с динамической памятью:
  - оператор new;
  - оператор delete.
- Понятие ссылки &:
  - что такое ссылка;
  - синтаксис объявления ссылки;
  - синтаксис объявления const ссылки;
  - примеры.
- Понятие &&.
- Передача аргументов внутрь функции:
  - передача по значению;
  - передача по ссылке;
  - передача по указателю.
- Указатель на функцию:
  - понятие адреса функции;
  - необходимость использования указателя на функцию;
  - синтаксис объявления указателя на функцию;
  - примеры использования указателя на функцию, массива указателей на функции.

Модуль 8

Строки

1. Строки.
   - Понятие строки в стиле “С” как массива символов.
   - Знакомство с нультерминированными строками.
   - Варианты инициализации строки при объявлении.
   - Тонкости ввода, вывода строк.

- Алгоритмы, используемые при работе со строками.
- Анализ функций из библиотеки string.h:
  - длина строки;
  - копирование строк;
  - конкатенация строк;
  - поиск символов в строке;
  - поиск подстроки в строке;
  - работа с различным регистром символов в строке;
  - замена символов и подстрок в строке.

Модуль 9

Многомерные динамические массивы

1. Указатель на указатель.
   - Необходимость использования указателя на указатель.
   - Синтаксис объявления указателя на указатель.
   - Примеры использования.
   - Использование указателя на указатель для передачи указателя внутрь функции.
2. Многомерные динамические массивы.
   - Необходимость использования многомерного динамического массива.
   - Двумерный массив как частный случай многомерного массива.
   - Схема расположения двумерного динамического массива в оперативной памяти.
   - Примеры использования двумерного динамического массива.

Модуль 10

Структуры

1. Структуры.
   - Необходимость использования структур.
   - Синтаксис объявления структур.
   - Инициализация и доступ к элементам структуры.
   - Массивы структур.
   - Вложенные структуры.
   - Структуры как аргументы функций.

- Ключевое слово typedef.
- Применение typedef для структур.

2. Краткие сведения из курса двоичной арифметики.
   - Что такое системы исчисления.
   - Какие бывают системы исчисления.
   - Что такое двоичная, восьмеричная, шестнадцатеричная система исчисления.
   - Использование нескольких систем исчисления.
   - Арифметические операции в разных системах исчисления.
   - Арифметические операции в двоичной системе исчисления.
3. Объединения.
   - Необходимость использования объединений.
   - Синтаксис объявления объединений.
   - Примеры использования объединений.
4. Битовые поля.
   - Необходимость использования битовых полей.
   - Синтаксис объявления битовых полей.
   - Примеры использования битовых полей.

Модуль 11

Препроцессор в приложениях

1. Директивы препроцессора.
   - Понятие препроцессора.
   - Директивы препроцессора.
   - Препроцессорная директива #include.
   - Препроцессорная директива #define для создания констант.
   - Макроопределения с параметрами.
   - Условная компиляция:
     - понятие условной компиляции;
     - директива #if;
     - директива #if #else;
     - директива #ifdef;
     - директива #ifndef;
     - директива #undef.

2. Многофайловые проекты.
   - Что такое многофайловый проект.
   - Зачем нужно разделять проект на несколько файлов.
   - Как создать многофайловый проект.

Модуль 12

Использование файловой системы

1. Стандартная библиотека ввода-вывода в языке C. Функции scanf, printf.
2. Выделение и очистка памяти в языке C.
   - Функция malloc.
   - Функция calloc.
   - Функция realloc.
   - Функция free.
3. Перенаправление ввода-вывода.
4. Использование аргументов командной строки.
5. Работа с файлами.
   - Понятие файла.
   - Понятие дескриптора файла.
   - Текстовые и двоичные файлы.
   - Текстовый и двоичный режим открытия файлов.
   - Открытие файлов.
   - Сохранение данных в файл.
   - Чтение данных из файла.
   - Закрытие файла.
   - Понятие буфера при работе с файлами.
   - Текущая позиция в файле. Позиционирование по файлу.
6. Битовые операции.
   - Цели и задачи битовых операций.
   - Битовое «И».
   - Битовое «ИЛИ».
   - Битовое «ИСКЛЮЧАЮЩЕЕ ИЛИ».
   - Битовое отрицание.
   - Битовые сдвиги.

7. Поиск файлов.
   - Поиск файлов по заданной маске и пути.
   - Использование битовых операций при поиске файлов.

Модуль 13

Экзамен

ШАГ компьютерная АКАДЕМИЯ

Программа курса
«Объектно-ориентированное программирование на C++ и библиотека стандартных шаблонов (STL)

Об'єктно-орієнтоване програмування на C ++
і бібліотека стандартних шаблонів (STL)

Object-Oriented Programming Using C++
and Standard Template Library (STL)

Цель курса

Обучить слушателя разработке приложений с использованием объектно-ориентированного подхода, заложенного в язык программирования C++. Научить выбирать правильные механизмы для решения той или иной задачи. Ознакомить с тонкостями использования инкапсуляции, наследования, полиморфизма, динамических структур данных, библиотеки STL.

По окончании курса слушатель будет:

- понимать базовые и расширенные концепции ООП;
- реализовывать пользовательские конструкторы копирования;
- грамотно перегружать операторы методами-членами класса и внешними функциями;
- разбираться в тонкостях динамических структур данных;
- уметь проектировать иерархии классов;
- программировать с использованием шаблонов и виртуальных методов;
- использовать библиотеку стандартных шаблонов STL;
- перехватывать исключения и строить собственные иерархии пользовательских исключений;
- взаимодействовать с файловыми потоками и потоками данных.

По окончании данного курса студент сдает теоретический экзамен по материалам курса. Для допуска к экзамену, должны быть сданы все домашние и практические задания.

Перед началом данного предмета необходимо предоставить студентам доступ к следующим курсам Cisco Networking Academy:

- Programming Essentials in C++;
- Advanced Programming in C++.

Тематический план

Модуль 1. Введение в объектно-ориентированное программирование на C++

Модуль 2. Указатель this и конструктор копирования

Модуль 3. Константные методы, explicit конструктора

Модуль 4. Перегрузка операторов

Модуль 5. Шаблоны классов, класс string

Модуль 6. Динамические структуры данных

Модуль 7. Агрегация, композиция и наследование

Модуль 8. Виртуальные методы

Модуль 9. Обработка исключительных ситуаций

Модуль 10. Пространства имен

Модуль 11. Преобразования типов в C++

Модуль 12. Работа с потоками в языке C++

Модуль 13. Умные указатели, работа со стандартной библиотекой C++, лямбда-функции

Модуль 14. Системы контроля версий

Модуль 15. Экзамен

Модуль 1

Введение в объектно-ориентированное программирование на C++

1. Вступление.
2. История и этапы развития языка C++.
3. Сравнительный анализ языка C++ с другими языками программирования (C, PASCAL, BASIC).
4. Три принципа объектно-ориентированного программирования.
   - Инкапсуляция. Определение, примеры использования в повседневной среде.
   - Полиморфизм. Определение, примеры использования в повседневной среде.
   - Наследование. Определение, примеры использования в повседневной среде.
5. Класс и объект.
6. Классы.
   - Понятие класса.
   - Синтаксис объявления.
   - Спецификаторы доступа:
     - public;
     - private;
     - protected.
7. Переменные-члены класса.
8. Методы-члены.
   - Реализация тела метода внутри класса.
   - Вынос тела метода за класс.
9. Практические примеры работы с классами.
   - Использование спецификаторов доступа.
   - Реализация практических примеров (Студент, Прямоугольник, Точка, Машина и так далее).
10. Понятие аксессора, инспектора, модификатора.
    - Определение.
    - Реализация.

11. Встроенные (inline) методы в классах.
    - Необходимость использования.
    - Примеры объявления и использования.
    - Ограничения при использовании inline методов.
12. Сравнительный анализ структур и классов.
13. Конструктор.
    - Проблемы, возникающие при использовании неинициализированных переменных.
    - Понятие конструктора.
    - Синтаксис объявления.
    - Конструктор по умолчанию.
    - Конструктор, принимающий параметры.
    - Перегруженные конструкторы.
    - Примеры использования (например: классы Студент, Точка, Машина и так далее).
14. Деструктор.
    - Утечки ресурсов. Причины их возникновения и плачевные последствия данного явления.
    - Понятие деструктора.
    - Синтаксис объявления.
    - Примеры использования (например: классы Студент, Массив, Строка и так далее).
15. Указатели на объекты.
16. Массивы объектов.
17. Инициализаторы.
    - Синтаксис объявления.
    - Примеры практического использования (инициализация поля класса, константы члена класса, инициализация внутреннего объекта).
18. Унифицированная инициализация объектов.
19. Инициализация членов класса.
20. Делегирование конструкторов.
21. Статические переменные-члены и статические функции-члены класса.
    - Необходимость использования статических членов (показать на практическом примере, например: подсчет количества объектов и так далее).
    - Синтаксис объявления статических переменных-членов класса.

- Синтаксис объявления статических функций-членов класса.
- Отличие статистических функций-членов класса от функций-членов класса.

Модуль 2

Указатель this и конструктор копирования

1. Указатель this.
   - Понятие указателя this.
   - Практические примеры использования указателя this.
2. Конструктор копирования.
   - Понятие побитового копирования.
   - Проблемы, связанные с побитовым копированием.
   - Проблемные ситуации, требующие конструктора копирования (передача по значению объекта, возврат объекта по значению, создание объекта в форме присваивания другого объекта).
   - Синтаксис конструктора копирования.
   - Примеры использования конструктора копирования (классы Вектор, Строка, Матрица и так далее):
     - обсуждение тонкостей конструктора копирования;
     - спецификатор const;
     - необходимость передачи по ссылке.

Модуль 3

Константные методы, explicit конструкторы

1. Константный метод.
   - Синтаксис объявления.
   - Особенности указателя this в константном методе.
   - Примеры использования.
2. Объявление конструктора с использованием ключевого слова explicit.
   - Примеры ситуации, иллюстрирующие неявное создание объекта.
   - Ключевое слово explicit и его использование.
   - Объявление конструктора с использованием ключевого слова explicit.

Модуль 4

Перегрузка операторов

1. Необходимость использования перегрузки операторов.
   - Примеры кода (реализация классов, например таких как дробь, матрица, через обычные методы члены типа Sum, Mult и так далее).
   - Логичность использования стандартных символов (+, – , >,< и так далее).
2. Перегрузка операторов.
   - Общие понятия перегрузки операторов:
     - классификация операторов на основании количества операндов (бинарные, унарные, триадный);
     - определение перегрузки операторов;
     - различные виды перегрузки (метод-член, функция-друг, глобальная функция).
   - Синтаксис перегрузки операторов методом – членом (унарный, бинарный вид).
   - Примеры перегрузки операторов:
     - перегрузка арифметических операторов:
       - перегрузка операторов +,-, * и так далее;
       - перегрузка инкремента и декремента:
         - цели и задачи перегрузки инкремента и декремента;
         - синтаксис перегрузки;
         - отличия перегрузки постфиксной и префиксной формы.
     - перегрузка логических операторов;
     - возврат по ссылке;
     - перегрузка оператора присваивания.
3. Конструктор переноса.
   - Что такое конструктор переноса.
   - Цели и задачи конструктора переноса.
   - Примеры реализации.
4. Применение переноса при перегрузке оператора присваивания.
5. Заданные по умолчанию методы (default) и удаленные методы (delete).
6. Специальные перегрузки.
   - Перегрузка [].
   - Перегрузка ().
   - Перегрузка оператора преобразования типов.
   - Использование explicit для преобразований, определяемых классом.

7. Список операторов, которые невозможно перегрузить.
8. Статический полиморфизм и перегрузка операторов как частный случай.
9. Перегрузка операторов дружественными и глобальными функциями.
   - Перегрузка операторов глобальными функциями:
     - отличия синтаксиса;
     - примеры использования (классы Вектор, Матрица, Строка и так далее).
   - Дружественные функции:
     - понятие дружественной функции;
     - цели и задачи дружественных функций;
     - ключевое слово friend;
     - отличия дружественных функций от методов класса;
     - примеры использования дружественных функций;
     - перегрузка операторов с использованием дружественных функций;
     - список операторов, которые невозможно перегрузить не методами-членами классов.
   - Перегрузка ввода-вывода:
     - потоковые классы ostream и istream;
     - синтаксис перегрузки ввода-вывода;
     - примеры использования (классы Вектор, Матрица, Строка и так далее).
   - Дружественные классы:
     - цели и задачи;
     - синтаксис и примеры использования.

Модуль 5

Шаблоны классов, класс string

1. Статический полиморфизм и шаблоны как частный случай.
2. Шаблоны классов.
   - Шаблоны классов.
   - Полная специализация.
   - Частичная специализация.
   - Примеры создания шаблонов классов (например: Вектор, Матрица и так далее).
3. Шаблоны с переменным числом аргументов.
4. Использование std::initializer_list.

5. Класс string.
   - Что такое string.
   - Цели и задачи класса string.
   - Анализ устройства класса string.
   - Примеры использования класса string.

Модуль 6

Динамические структуры данных

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.
   - Списки:
     - понятие списка;
     - односвязный список:
       - добавление элементов в список;
       - обход списка;
       - удаление элементов;
       - замена элементов;
       - показ элементов списка;
       - поиск элемента в списке;
       - примеры создания и использования списков.
     - Двусвязный список:
       - добавление элементов в список;
       - обход списка;
       - удаление элементов;
       - замена элементов;
       - показ элементов списка;

        - поиск элемента в списке;
        - примеры создания и использования списков;
        - сравнительный анализ типов списка.
    - Деревья:
        - понятие дерева;
        - бинарное дерево поиска;
        - сортирующее дерево;
        - красно-черное дерево;
        - операции, выполняемые над деревом:
            - добавление элемента;
            - получение значения элемента;
            - удаление элемента;
            - показ дерева;
            - поиск элемента;
            - уничтожение дерева;
            - примеры создания и использования бинарных деревьев поиска, сортирующих деревьев, красно-черных деревьев.
    - Сравнительный анализ изученных динамических структур данных.

Модуль 7

Агрегация, композиция и наследование

1. Вложенный класс.
    - Синтаксис объявления.
    - Цели и задачи вложенных классов.
    - Примеры использования (например: связанный список и так далее).
2. Агрегация и композиция.
    - Понятие агрегации.
    - Понятие композиции.
    - Отличие агрегации от композиции.
3. Наследование.
    - Цели и задачи наследования.
    - Примеры использования наследования в окружающей среде.
    - Типы наследования.
    - Понятия базового и дочернего класса.
    - Одиночное наследование:
        - синтаксис одиночного наследования;

- спецификатор доступа protected;
- спецификаторы доступа при одиночном наследовании;
- поведение конструкторов и деструкторов при одиночном наследовании;
- примеры использования одиночного наследования (например: иерархии Человек-Студент, Человек-Милиционер и так далее).

- Множественное наследование:
  - синтаксис множественного наследования;
  - спецификаторы доступа при множественном наследовании;
  - поведение конструкторов и деструкторов при множественном наследовании;
  - примеры использования множественного наследования;
  - недостатки использования множественного наследования.
- Обсуждение плюсов и минусов наследования.
- Наследование шаблонов:
  - виртуальный базовый класс;
  - пример проблемы ромба;
  - спецификатор virtual и виртуальное наследование;
  - пример использования виртуального базового класса.

Модуль 8

Виртуальные методы

1. Указатель на базовый класс.
2. Виртуальные методы.
3. Ранее и позднее связывание.
4. Статический и динамический полиморфизм.
5. Таблица виртуальных функций.
6. Использование спецификаторов override и final.
7. Примеры использования виртуальных методов.
8. Абстрактный класс.
   - Чисто виртуальный метод.
   - Абстрактный класс.
9. Виртуальный деструктор.
10. Чисто виртуальный деструктор.

Модуль 9

Обработка исключительных ситуаций

1. Понятие исключительной ситуации.
2. Необходимость обработки исключительных ситуаций.
3. Типы исключительных ситуаций.
4. Базовые понятие обработки исключительных ситуаций.
   - Ключевое слово try.
   - Ключевое слово catch.
   - Ключевое слово throw.
   - Примеры использования обработки исключительных ситуаций.
5. Понятие необработанного исключения.
6. Специальная форма catch(…).
7. Исключения и функции.
8. Описание списка исключений генерируемых функцией.
9. Раскрутка стека вызовов.
10. Повторная генерация исключения.
11. Построение иерархии пользовательских классов исключений.
12. Стандартный класс exception и его потомки.
13. Обработка ошибок при выделении памяти.
14. Обработка не пойманных и неожидаемых исключений.

Модуль 10

Пространства имен

1. Причины возникновения пространств имен.
2. Синтаксис объявления.
3. Оператор using.
4. Вложенные пространства.
5. Тонкости использования пространств имен.

Модуль 11

Преобразования типов в C++

1. Оператор typeid.
2. Преобразования типов в C++.
   - dynamic_cast.
   - static_cast.
   - reinterept_cast.
   - const_cast.

Модуль 12

Работа с потоками в языке C++

1. Понятие потока.
2. Виды потоков.
3. Ввод и вывод в языке C++.
4. Файловый ввод-вывод в C++.
   - Класс ofstream.
   - Класс ifstream.
   - Класс fstream.
5. Файловые операции.
   - Открытие файла.
   - Закрытие файла.
   - Чтение данных.
   - Запись данных.
   - Позиционирование по файлу.
   - Перегрузка <<,>> для чтения, сохранения данных в файл.

Модуль 13

Умные указатели, работа со стандартной библиотекой C++, лямбда-функции

1. Умные указатели.
   - Что такое умный указатель.
   - Классы умных указателей:
     - auto_ptr;

- shared_ptr;
- unique_ptr;
- концептуальные отличия классов умных указателей.

- Особенности использования auto_ptr.
- Особенности использования unique_ptr.
- Особенности использования shared_ptr.

2. Стандартная библиотека шаблонов (STL).
   - Что такое STL.
   - История возникновения STL.
   - Цели и задачи стандартной библиотеки шаблонов.
3. Основные понятия STL.
   - Контейнер.
   - Итератор.
   - Алгоритм.
   - Функтор.
4. Контейнер.
   - Что такое контейнер.
   - Типы контейнеров.
   - Пример использования контейнера vector.
5. Итератор.
   - Что такое итератор.
   - Типы итераторов.
   - Почему так много типов итераторов.
   - Пример использования итераторов.
6. Подробно о контейнерах.
7. Анализ и использование классов list, map, multimap.
8. Практические примеры использования классов контейнеров.
9. Использование функторов.
10. Использование алгоритмов.
11. Практические примеры использования функторов, алгоритмов.
12. Лямбда-функции.
    - Что такое лямбда-функция.
    - Цели и задачи лямбда-функций.
    - Примеры использования.

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.
   - 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.
   - Использование внешних сервисов (github).

Модуль 15

Экзамен

Программа курса
«Объектно-ориентированное проектирование. Язык UML»

Об'єктно-орієнтоване проектування. Мова UML
Object-Oriented Design Using UML

Цель курса

Обучить слушателя принципам объектно-ориентированного проектирования. Научить использовать язык UML для построения диаграмм, охватывающих различные аспекты приложений.

По окончании курса слушатель будет:

- понимать концепции объектно-ориентированного проектирования;
- уметь читать и анализировать диаграммы UML;
- уметь проектировать диаграммы классов, состояний, деятельности и другие;
- использовать необходимый инструментарий для построения диаграмм;
- разбираться в тонкостях языка UML;
- уметь проектировать иерархии классов на основании ООП;
- выполнять декомпозицию.

Этот курс находится в начале специализации для того, чтобы сформировать у студента правильное мировоззрение и принципы для разработки качественного программного обеспечения. Курс UML носит достаточно абстрактный характер, поэтому преподавателю нужно максимально подробно объяснять концепции и приводить большое количество практических примеров.

По окончании данного курса студент сдает практический и теоретический экзамен по материалам курса. Для допуска к экзамену, должны быть сданы все домашние и практические задания. В качестве практической части экзамена, студент реализовывает комплекс диаграмм для некоторого приложения.

Тематический план

Модуль 1. Введение в объектно-ориентированный анализ и проектирование

Модуль 2. Диаграммы в UML

Модуль 3. Диаграммы вариантов использования

Модуль 4. Диаграмма классов

Модуль 5. Диаграмма состояний, диаграмма деятельности

Модуль 6. Диаграмма последовательности, диаграмма кооперации

Модуль 7. Диаграмма компонентов и диаграмма развертывания

Модуль 8. Экзамен

Модуль 1

Введение в объектно-ориентированный анализ и проектирование

1. Введение в предметную область.
2. Сложности при разработке программного обеспечения.
3. Причины возникновения ООП.
4. Базовые понятия.
   - Декомпозиция.
   - Абстракция.
   - Модульность.
   - Иерархия.
5. Обзор существующих методологий.
   - Методология процедурно-ориентированного программирования:
     - исторический экскурс;
     - процедурно-ориентированные языки;
     - анализ процедурно-ориентированной методологии.
   - Методология объектно-ориентированного программирования:
     - исторический экскурс;
     - фундаментальные понятия объектно-ориентированного программирования;
     - объектно-ориентированные языки программирования;
     - анализ объектно-ориентированной методологии.
   - Методология объектно-ориентированного анализа и проектирования:
     - исторический экскурс;
     - язык UML.
   - Методология системного анализа и системного моделирования:
     - исторический экскурс.
6. Классы и объекты.
   - Понятие класса и объекта.
   - Объект:
     - природа объектов;
     - состояние объектов;
     - поведение объектов;

- объекты как автоматы;
- идентичность;
- отношения между объектами;
- типы отношений:
  - связи:
    - актер;
    - сервер;
    - агент.
  - агрегация.

- Класс:
  - интерфейс класса;
  - реализация класса;
  - отношения между классами;
  - наследование;
  - инстанционирование;
  - метаклассы.
- Отношения между классами и объектами.

7. Экскурс в диаграммы.
   - Диаграммы «сущность-связь».
   - Диаграммы функционального моделирования.
   - Диаграммы потоков данных.
8. История развития языка UML.
   - Исторический экскурс в языки объектно-ориентированного моделирования:
     - метод Гради Буча Booch;
     - метод Джеймса Рамбо Object Modeling Technique;
     - метод Айвара Джекобсона Object-Oriented Software Engineering;
     - другие методы.
   - Унификация существующих методов:
     - появление языка UML (Unified Modeling Language);
     - версии языка UML.

Модуль 2

Диаграммы в UML

1. Понятие диаграммы в UML.
2. Краткий обзор и анализ существующих видов диаграмм.

- Диаграмма вариантов использования (use case diagram).
- Диаграмма классов (class diagram).
- Диаграмма состояний (statechart diagram).
- Диаграмма деятельности (activity diagram).
- Диаграмма последовательности (sequence diagram).
- Диаграмма кооперации (collaboration diagram).
- Диаграмма компонентов (component diagram).
- Диаграмма развертывания (deployment diagram).

3. Инструментарий для построения диаграмм.
   - Microsoft Visio.
   - ArgoUML.
   - StarUML.
   - Rational Rose.
   - Другие инструментальные средства.

Модуль 3

Диаграммы вариантов использования

1. Цели данного типа диаграмм.
2. Базовые понятия.
   - Вариант использования.
   - Актер или действующее лицо.
   - Интерфейс.
   - Примечания.
3. Отношения.
   - Отношение ассоциации (association relationship).
   - Отношение расширения (extend relationship).
   - Отношение обобщения (generalization relationship).
   - Отношение включения (include relationship).
   - Сравнительный анализ отношений.
4. Практические примеры построения диаграмм вариантов использования. Например такие как: система продажи/покупки товаров с использованием электронного магазина или система бронирования железнодорожных билетов и так далее.

Модуль 4

Диаграмма классов

1. Цели данного типа диаграмм.
2. Базовые понятия.
   - Класс:
     - операция;
     - атрибут.
   - Объект.
   - Интерфейс.
   - Шаблон.
3. Отношения между классами.
   - Отношение зависимости (dependency relationship).
   - Отношение ассоциации (association relationship).
   - Отношение обобщения (generalization relationship).
   - Отношение реализации (realization relationship).
   - Сравнительный анализ отношений.
4. Практические примеры построения диаграмм классов.

Модуль 5

Диаграмма состояний, диаграмма деятельности

1. Цели данного типа диаграмм.
2. Базовые понятия.
   - Автоматы.
   - Состояние:
     - понятие состояния;
     - имя состояния;
     - список внутренних действий;
     - начальное состояние;
     - конечное состояние.
   - Переход:
     - понятие перехода;
     - событие;
     - сторожевое условие;
     - выражение действия.

- Составное состояние и подсостояние:
  - последовательные подсостояния;
  - параллельные подсостояния.
- Историческое состояние.
- Сложные переходы:
  - переходы между параллельными состояниями;
  - переходы между составными состояниями;
  - синхронизирующие состояния.

3. Практические примеры построения диаграмм состояний.
4. Цели данного типа диаграмм.
5. Базовые понятия.
   - Состояние действия.
   - Переходы.
   - Дорожки.
   - Объекты.
6. Практические примеры построения диаграмм деятельности.

Модуль 6

Диаграмма последовательности, диаграмма кооперации

1. Цели данного типа диаграмм.
2. Базовые понятия.
   - Объекты:
     - линия жизни объекта;
     - фокус управления.
   - Сообщения:
     - ветвление потока управления;
     - стереотипы сообщений;
     - временные ограничения на диаграммах последовательности;
     - комментарии или примечания.
3. Практические примеры построения диаграмм последовательности.
4. Цели данного типа диаграмм.
5. Базовые понятия.
   - Кооперация.
   - Диаграмма кооперации уровня спецификации.

- Объекты:
  - мультиобъект;
  - активный объект;
  - составной объект.
- Связи.
- Стереотипы связей.
- Сообщения.
- Формат записи сообщений.

6. Практические примеры построения диаграмм кооперации.

Модуль 7

Диаграмма компонентов и диаграмма развертывания

1. Цели данного типа диаграмм.
2. Базовые понятия.
   - Компонент:
     - понятие компонента;
     - имя компонента;
     - виды компонентов.
   - Интерфейс.
   - Зависимости.
3. Практические примеры построения диаграмм компонентов.
4. Цели данного типа диаграмм.
5. Базовые понятия.
   - Узел.
   - Соединения.
6. Практические примеры построения диаграмм развертывания.

Модуль 8

Экзамен

Программа курса
«Паттерны проектирования»

Паттерни проектування
Design Patterns

Цель курса

Обучить слушателя разработке приложений с использованием паттернов проектирования. Научить применять правильные паттерны для решения той или иной задачи. Ознакомить с тонкостями использования теоретического и практического применения паттернов.

По окончании курса слушатель будет:

- понимать причины возникновения паттернов;
- понимать понятие паттерн проектирования;
- разбираться в категориях паттернов;
- уметь отличать паттерны одной категории;
- правильно выбирать и применять паттерн для решения той или иной задачи;
- уметь проектировать классы с учетом принципов SOLID;
- правильно использовать паттерн MVC.

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.

Экзаменационное практическое задание должно охватывать изученные паттерны из разных категорий.

Тематический план

Модуль 1. Введение в паттерны проектирования

Модуль 2. Порождающие паттерны

Модуль 3. Структурные паттерны

Модуль 4. Паттерны поведения

Модуль 5. Паттерн MVC

Модуль 6. Принципы проектирования классов SOLID

Модуль 7. Экзамен

Модуль 1

Введение в паттерны проектирования

1. Анализ существующих тенденций в программном обеспечении.
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. Использование UML при анализе паттернов проектирования.
   - Диаграмма классов.
   - Диаграмма объектов.
   - Диаграмма взаимодействия.

Модуль 2

Порождающие паттерны

1. Понятие порождающего паттерна.
2. Abstract Factory.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
3. Builder.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
4. Factory Method.
   - Цель паттерна.
   - Причины возникновения паттерна.

- Структура паттерна.
- Результаты использования паттерна.
- Практический пример использования паттерна.

5. Prototype.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
6. Singleton.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
7. Анализ и сравнение порождающих паттернов.
8. Практические примеры использования порождающих паттернов.

Модуль 3

Структурные паттерны

1. Понятие структурного паттерна.
2. Adapter.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
3. Bridge.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.

4. Composite.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
5. Decorator.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
6. Facade.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
7. Flyweight.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
8. Proxy.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
9. Анализ и сравнение структурных паттернов.
10. Практические примеры использования структурных паттернов.

Модуль 4

Паттерны поведения

1. Понятие паттерна поведения.
2. Chain of Responsibility.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
3. Command.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
4. Interpreter.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
5. Iterator.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
6. Mediator.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.

7. Memento.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
8. Observer.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
9. State.
   - Цель паттерна.
   - Причины возникновения паттерна.
   - Структура паттерна.
   - Результаты использования паттерна.
   - Практический пример использования паттерна.
10. Strategy.
    - Цель паттерна.
    - Причины возникновения паттерна.
    - Структура паттерна.
    - Результаты использования паттерна.
    - Практический пример использования паттерна.
11. Template Method.
    - Цель паттерна.
    - Причины возникновения паттерна.
    - Структура паттерна.
    - Результаты использования паттерна.
    - Практический пример использования паттерна.
12. Visitor.
    - Цель паттерна.
    - Причины возникновения паттерна.
    - Структура паттерна.
    - Результаты использования паттерна.
    - Практический пример использования паттерна.

13. Анализ и сравнение паттернов поведения.
14. Практические примеры использования паттернов поведения.

Модуль 5

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 6

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 7

Экзамен

ШАГ компьютерная АКАДЕМИЯ

Программа курса «Платформа Microsoft .NET и язык программирования C#»

Платформа Microsoft .NET та мова програмування C#
Microsoft .NET Framework and C# Programming Language

Цель курса

Обучить слушателя основам разработки приложений с использованием платформы Microsoft. NET и языка программирования C#. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

По окончании курса слушатель будет:

- понимать причины возникновения платформы Microsoft. NET;
- оперировать базовыми терминами платформы Microsoft. NET: CLR, CLS, CTS, BCL;
- уметь использовать рефлекторы и дотфускаторы;
- разбираться в тонкостях реализации ООП в C#;
- уметь создавать классы пользовательских исключений;
- уметь создавать пользовательские делегаты и события;
- взаимодействовать со сборщиком мусора;
- уметь использовать механизмы сериализации;
- знать основы использования LINQ.

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.

Перед началом данного предмета необходимо предоставить студентам доступ к следующим курсам Microsoft Imagine Academy:

- Working with Non-Relational Data.

Тематический план

Модуль 1. Введение в платформу Microsoft. NET

Модуль 2. Основы языка программирования C#, массивы и строкиы

Модуль 3. Введение в классы

Модуль 4. Обработка исключений

Модуль 5. Пространства имен

Модуль 6. Перегрузка операторов

Модуль 7. Индексаторы и свойства

Модуль 8. Наследование

Модуль 9. Интерфейсы

Модуль 10. Структуры, перечисления

Модуль 11. Делегаты, события

Модуль 12. Введение в Generics

Модуль 13. Сборка мусора

Модуль 14. Взаимодействие с файловой системой

Модуль 15. Основы XML

Модуль 16. Введение в LINQ

Модуль 17. Сериализация объектов

Модуль 18. Логирование

Модуль 19. Экзамен

Модуль 1

Введение в платформу Microsoft. NET

1. Введение в платформу Microsoft. NET:
   - история и этапы развития технологий программирования;
   - причины возникновения платформы Microsoft. NET;
   - сравнительный анализ преимуществ и недостатков платформы Microsoft. NET.
2. Базовые понятия платформы Microsoft. NET:
   - архитектура платформы Microsoft. NET;
   - общеязыковая среда исполнения CLR (common language runtime);
   - стандартная система типов CTS (common type system);
   - стандартная языковая спецификация CLS (common language specification);
   - библиотека классов FCL (BCL);
   - языки платформы Microsoft. NET;
   - схема компиляции и исполнения приложения платформы Microsoft. NET;
   - язык MSIL (Microsoft Intermediate Language);
   - понятия метаданных, манифеста, сборки.
3. Введение в язык программирования C#:
   - плюсы и минусы языка программирования C#;
   - простейшая программа на языке программирования C#.
4. Рефлекторы и дотфускаторы:
   - что такое рефлектор;
   - необходимость использования рефлектора;
   - обзор существующих рефлекторов;
   - что такое дотфускатор;
   - необходимость использования дотфускаторов;
   - обзор существующих дотфускаторов.

Модуль 2

Основы языка программирования C#, массивы и строки

1. Типы данных:
   - целочисленные типы данных;
   - типы данных для чисел с плавающей точкой;
   - символьный тип данных;
   - другие типы данных.
2. Литералы.
3. Переменные:
   - понятие переменной;
   - правила именования переменных;
   - область видимости переменных.
4. Анонимные типы.
5. Ввод, вывод в консольном приложении.
6. Структурные и ссылочные типы.
7. Преобразование типов:
   - явное преобразование;
   - неявное преобразование.
8. Операторы:
   - арифметические операторы;
   - операторы отношений;
   - логические операторы;
   - битовые операторы;
   - оператор присваивания;
   - приоритет операторов;
   - null-conditional оператор.
9. Условия:
   - условный оператор if;
   - условный оператор if else;
   - условный оператор switch;
   - оператор ?.
10. Циклы:
    - цикл for;

- цикл while;
- цикл do while;
- цикл foreach;
- инструкция break;
- инструкция continue;
- инструкция goto.

11. Использование nameoff.
12. Массивы:
    - одномерные массивы;
    - многомерные массивы;
    - рваные массивы;
    - использование цикла foreach.
13. Строки:
    - создание строки;
    - операции со строками;
    - особенности использования строк;
14. Использование аргументов командной строки.

Модуль 3

Введение в классы

1. Синтаксис объявления класса.
2. Спецификаторы доступа языка программирования C#.
3. Поля класса.
4. Методы класса:
    - передача параметров;
    - ключевое слово return;
    - перегрузка методов;
    - краткий синтаксис однострочных методов;
    - реализация тела метода в виде выражения.
5. Конструкторы:
    - понятие конструктора;
    - параметризованный конструктор;
    - перегруженные конструкторы;
    - статические конструкторы.

6. Ключевое слово this.
7. Использование ref и out параметров:
   - использование модификатора ref;
   - использование модификатора out;
   - кортежи.
8. Создание методов с переменным количеством аргументов.
9. Частичные типы (partial types).

Модуль 4

Обработка исключений

1. Иерархия исключений:
   - базовый класс system.exception;
   - анализ иерархии стандартных исключений.
2. Основы обработки исключений:
   - ключевое слово try;
   - ключевое слово catch;
   - ключевое слово throw;
   - ключевое слово finally.
3. Тонкости обработки исключений.
   - перехват всех исключений;
   - вложенные блоки try;
   - повторное генерирование исключений.
4. Применение конструкций checked и unchecked.
5. Фильтры исключений.

Модуль 5

Пространства имен

1. Что такое пространство имен?
2. Цели и задачи пространства имен.
3. Ключевое слово using.
4. Объявление пространства имен.
5. Вложенные пространства имен.

6. Разбиение пространства имен на части.
7. Пространство имен по умолчанию.
8. Вторая форма using.
9. Использование using для подключения статических членов.

Модуль 6

Перегрузка операторов

1. Введение в перегрузку операторов.
2. Перегрузка унарных операторов.
3. Перегрузка бинарных операторов.
4. Перегрузка операторов отношений.
5. Перегрузка логических операторов.
6. Перегрузка операторов true и false.
7. Перегрузка операторов преобразования.

Модуль 7

Индексаторы и свойства

1. Индексаторы:
   - понятие индексатора;
   - создание одномерных индексаторов;
   - создание многомерных индексаторов;
   - перегрузка индексаторов.
2. Свойства:
   - что такое свойства;
   - синтаксис объявления свойств;
   - краткий синтаксис однострочных свойств;
   - примеры использования свойств;
   - автоматические свойства (auto-property):
     - что такое автоматические свойства;
     - примеры использования автоматических свойств;
     - инициализация автоматических свойств.

Модуль 8

Наследование

1. Наследование в C#:
   - анализ механизма наследования в C#;
   - спецификаторы доступа при наследовании;
   - особенности использования конструкторов при наследовании;
   - сокрытие имен при наследовании;
   - ключевое слово base;
   - наследование и исключения;
   - наследование от стандартных классов исключений.
2. Использование ключевого слова sealed.
3. Использование ссылок на базовый класс.
4. Виртуальные методы
   - что такое виртуальный метод;
   - необходимость использования виртуальных методов;
   - переопределение виртуальных методов.
5. Абстрактный класс.
6. Анализ базового класса Object.
7. Упаковка, распаковка (boxing, unboxing).

Модуль 9

Интерфейсы

1. Понятие интерфейса.
2. Синтаксис объявления интерфейсов.
3. Примеры создания интерфейсов.
4. Интерфейсные ссылки.
5. Интерфейсные индексаторы, свойства.
6. Наследование интерфейсов.
7. Проблемы сокрытия имен при наследовании интерфейсов.
8. Анализ стандартных интерфейсов.

Модуль 10

Структуры, перечисления

1. Структуры:
   - понятие структуры;
   - синтаксис объявления структуры;
   - необходимость и особенности применения структур;
   - конструктор без параметров и структуры.
2. Перечисления (enum):
   - понятие перечисления;
   - синтаксис объявления перечисления;
   - необходимость и особенности применения перечисления;
   - установка базового типа перечисления;
   - использование методов для перечислений.
3. Nullable типы:
   - что такое Nullable тип;
   - цели и задачи Nullable типов;
   - операции доступные для Nullable типов;
   - примеры использования.

Модуль 11

Делегаты, события

1. Делегаты:
   - понятие делегата;
   - синтаксис объявления делегата;
   - цели и задачи делегатов;
   - вызов нескольких методов через делегат (multicasting);
   - базовые классы для делегатов:
     - system.Delegate;
     - system.MulticastDelegate.
2. События:
   - понятие события;
   - синтаксис объявления события;
   - необходимость и особенности применения событий;

- применение события для многоадресатного делегата;
- использование событийных средств доступа.

3. Анонимные методы.
4. Лямбда выражения.
5. Extension методы.

Модуль 12

Введение в Generics

1. Generics:
   - что такое generics;
   - необходимость использования generics;
   - создание generic классов;
   - вложенные типы внутрь generic класса;
   - использование ограничений;
   - создание generic интерфейсов;
   - создание generic делегатов;
   - создание generic методов.
2. Итераторы:
   - что такое итератор;
   - синтаксис и примеры использования итераторов.
3. Коллекции:
   - понятие коллекции;
   - generic коллекции;
   - классы коллекций List<T>, Dictionary<TKey,TValue>, Stack<T>, Queue<T>, SortedList<TKey,TValue> и другие;
   - интерфейсы коллекций IList<T>, IEnumerator<T>, IEnumerable<T>, ICollection<T>, IDictionary<TKey,TValue>, IComparer<T>, ...;
   - примеры использования классов generic коллекций.

Модуль 13

Сборка мусора

1. Жизненный цикл объектов.
2. Понятие сборщика мусора.

3. Деструктор и метод Finalize.
4. Метод Dispose и интерфейс IDisposable.
5. Класс System.GC.
6. Понятие поколений при сборке мусора.

Модуль 14

Взаимодействие с файловой системой

1. Модель потоков в C#. Пространство System.IO.
2. Класс Stream.
3. Анализ байтовых классов потоков.
4. Анализ символьных классов потоков.
5. Анализ двоичных классов потоков.
6. Использование класса FileStream для файловых операций.
7. Использование класса StreamWriter для файловых операций.
8. Использование класса StreamReader для файловых операций.
9. Использование класса BinaryWriter для файловых операций.
10. Использование класса BinaryReader для файловых операций.
11. Использование классов Directory, DirectoryInfo, FileInfo для файловых операций.
12. Регулярные выражения.

Модуль 15

Основы XML

1. Что такое XML?
2. История возникновения XML.
3. Цели и задачи XML.
4. XML-документ.
5. Синтаксис и структура XML-документа.
6. Описание структуры XML-документа с помощью DTD.
7. Пространства имен XML.
8. Понятие схемы, отличия схем от DTD.

9. Парсеры XML:
   - что такое парсер;
   - цели и задачи парсера;
   - DOM- и SAX-парсеры.
10. Примеры создания XML-документов.
11. XML-документация:
    - что такое XML-документация кода;
    - зачем использовать XML-документацию;
    - примеры использования.

Модуль 16

Введение в LINQ

1. Что такое LINQ?
2. Цели и задачи LINQ.
3. Понятие запроса:
   - запрос в LINQ;
   - синтаксис запроса;
   - исполнение запроса;
   - сортировка;
   - группировка;
   - другие операции.
4. Использование LINQ и коллекций.
5. Использование LINQ и XML.
6. Примеры использования.

Модуль 17

Сериализация объектов

1. Понятие атрибутов.
2. Что такое сериализация?
3. Отношения между объектами.
4. Графы отношений объектов.
5. Атрибуты для сериализации [Serializable] и [NonSerialized].

6. Форматы сериализации:
   - пространство System.Runtime.Serialization.Formatters;
   - двоичное форматирование. Класс BinaryFormatter;
   - soap форматирование. Класс SoapFormatter;
   - примеры использования сериализации;
   - создание пользовательского формата сериализации. Интерфейс ISerializable.

Модуль 18

Логирование

1. Что такое логирование?
2. Цели и задачи логирования.
3. Ситуации, требующие логирования.
4. Инструменты логирования:
   - log4net;
   - Serilog;
   - NLog.
5. Практические примеры использования.

Модуль 19

Экзамен

Программа курса
«Основы разработки приложений с использованием Windows Forms»

Основи розробки додатків з використанням Windows Forms
Basics of Application Development Using Windows Forms

Цель курса

Обучить слушателя разработке Windows-приложений с использованием платформы Microsoft .NET, языка программирования C# и библиотеки Windows Forms.
Исследовать стандартные и расширенные элементы управления Windows для организации пользовательского интерфейса.

По окончании курса слушатель будет:

- уметь создавать UI-интерфейс с помощью Windows Forms;
- совершенствовать средства пользовательского интерфейса приложения путем добавления динамических меню, графических строк состояния, диалоговых окон и различных панелей инструментов;
- использовать редактор ресурсов и другие средства, поставляемые в комплекте Visual Studio;
- использовать Windows Forms для разработки, создания и реализации в приложении меню, панелей инструментов, элементов диалоговых окон;
- использовать общие диалоги.

По окончании данного курса студент сдает практическое задание 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 Windows-программирование с использованием WinAPI и C++

Модуль 2. Введение в Windows Forms

Модуль 3. Взаимодействие с элементами управления

Модуль 4. Списки

Модуль 5. Прокрутка, индикаторы, ...

Модуль 6. Создание дополнительных форм

Модуль 7. Меню

Модуль 8. Использование расширенного текстового поля, дерева и списка

Модуль 9. Использование возможностей GDI+

Модуль 10. Экзамен

Модуль 1

Введение в Windows программирование с использованием WinAPI и C++

1. UNICODE.
   - Понятие кодировки.
   - Unicode кодировка.
   - Принципы работы с Unicode:
     - строковые функции для работы с Unicode строками;
     - функции для конвертирования Unicode строк:
       - wcstombs;
       - wmbstowcs;
       - MultiByteToWideChar;
       - WideCharToMultiByte.
     - макросы для работы с Unicode.
2. Программирование Windows-приложений. Введение.
3. Понятия многозадачность и многопоточность.
4. Независимость от аппаратных средств.
5. «Новые» типы данных.
6. События.
   - Понятие события.
   - Архитектура приложений, построенных на событиях.
7. Сообщение.
   - Понятие сообщения.
   - Соответствие события и сообщения.
   - Система сообщений Windows.
8. Очередь сообщений.
9. Окна.
   - Понятие окна.
   - Элемент управления как частный случай окна.
   - Дескриптор окна.
   - Стили окна.
   - Понятие оконного класса.

- Понятие оконной процедуры.
- Понятие оконного класса.

10. Утилита Spy++.
    - Цели и задачи утилиты Spy++.
    - Просмотр информации об окне (оконный класс, стили и так далее). В качестве примера можно привести «Блокнот», «Калькулятор» и так далее.
    - Просмотр очереди сообщений окна. В качестве примера можно привести «Блокнот», «Калькулятор» и так далее.
11. Минимальное Win32-приложение.
    - Написание приложения стандартного типа “Hello Step -) ”.
    - Функция WinMain.
    - Создание окна.
    - Регистрация оконного класса.
    - Обработка сообщений.
    - Оконная процедура.
    - Ресурсы.
12. Стили окна.
    - Стили окна.
    - Общие стили.
    - Расширенные стили.
    - Изменение стилей.
13. Окна сообщений.
14. Принципы обработки сообщений мыши.
15. Структуры RECT, POINT.
16. Таймер.
    - Понятие таймера.
    - Создание таймера.
    - Обработка событий таймера.
    - Остановка таймера.
17. Утилита ErrorLookup.
    - Понятие кода ошибки.
    - Получение кода ошибки.
    - Функция GetLastError.
    - Функция FormatMessage.

- Практический пример.
- Утилита GetLastError.

18. Практические задания.

Модуль 2

Введение в Windows Forms

1. Что такое Windows.Forms?
2. Отличие Windows.Forms от других GUI-библиотек.
3. Анализ типичного Windows Forms-приложения.
4. Окна сообщений.
5. Форма.
   - Понятие формы.
   - Свойства формы.
   - Модальные и немодальные формы.
6. Принципы обработки сообщений мыши.
7. Использование таймера.
8. Принципы работы со временем и датой.

Модуль 3

Взаимодействие с элементами управления

1. Элементы управления.
   - Что такое элемент управления.
   - Класс Control.
   - Общие принципы взаимодействия с элементами управления.
2. Статический текст. Класс Label.
3. Текстовое поле. Класс TextBox.
4. Кнопки.
   - Создание кнопок. Класс Button.
   - Создание переключателей. Класс RadioButton.
   - Создание селекторов. Класс CheckBox.
5. Элементы управления «Дата-Время» и «Календарь».
   - Класс DateTimePicker.
   - Класс MonthCalendar.

Модуль 4

Списки

1. Использование списков.
2. Типы списков.
3. Список. Класс ListBox.
4. Список с селекторами. Класс CheckedListBox.
5. Комбинированные списки. Класс ComboBox.

Модуль 5

Прокрутка, индикаторы, …

1. Полосы прокрутки. Классы VScrollBar, HScrollBar.
2. Индикатор. Класс ProgressBar.
3. Счетчик. Класс NumericUpDown.
4. Всплывающие подсказки. Класс ToolTip.
5. Строка состояния. Класс StatusStrip.
6. Слайдер. Класс TrackBar.
7. Использование системного трея.

Модуль 6

Создание дополнительных форм

1. Необходимость создания дополнительных форм.
2. Создание дополнительной формы.
3. Обмен данными между формами.
4. Создание немодальной формы.
5. Что такое общий диалог?
6. Типы общих диалогов.
7. Классы OpenFileDialog, SaveFileDialog.
8. Класс FolderBrowserDialog.

Модуль 7

Меню

1. Создание и использование меню.
2. Акселераторы.
3. Класс MenuStrip.
4. Создание меню на основе шаблона.
5. Динамическое создание меню.
6. Контекстное меню. Класс ContextMenuStrip.
7. Тулбар. Класс ToolStrip.

Модуль 8

Использование расширенного текстового поля, дерева и списка

1. Расширенное текстовое поле. Класс RichTextBox.
2. Дерево. Класс TreeView.
   - Добавление элементов дерева.
   - Удаление элементов дерева.
   - Проход по дереву.
3. Список. Класс ListView.
   - Добавление элементов списка.
   - Удаление элементов списка.
   - Проход по списку.
4. Использование механизма Drag and Drop.

Модуль 9

Использование возможностей GDI+

1. Что такое GDI+?
2. Пространство System.Drawing.
3. Графические примитивы в GDI+.
   - Перья.
   - Кисти.
   - Шрифты.

- Изображения.
- Регионы.
- Траектории.

4. Системы координат.
5. Класс Graphics.
   - Цели и задачи класса Graphics.
   - Способы получения доступа к объекту класса Graphics.
   - Общий анализ методов и свойств класса.
6. Событие Paint.
7. Методы для вывода простейших графических примитивов.
   - Отображение точки.
   - Отображение линии.
   - Отображение прямоугольника.
   - Отображение эллипса.
8. Структуры Color, Size, Rectangle, Point.
9. Кисти.
   - Типы кистей:
     - сплошная;
     - текстурная;
     - кисть с насечками;
     - градиентная кисть;
     - кисть с использованием траектории.
   - Пространство Drawing2D.
   - Класс Brush:
     - класс SolidBrush;
     - класс TextureBrush;
     - класс HatchBrush;
     - класс LinearGradientBrush;
     - класс PathGradientBrush.
10. Перо (карандаш).
    - Типы карандашей.
    - Класс Pen.
11. Примеры использования кистей и перьев.
12. Работа с изображениями.
    - Типы изображений.

- Класс Image:
  - класс Image;
  - класс Bitmap.
  - класс Metafile;
- Класс Bitmap:
  - загрузка изображений;
  - вывод изображений;
  - масштабирование изображений;
  - изменение изображений;
  - сохранение изображений.
- Примеры использования изображений.

Модуль 10

Экзамен

ШАГ компьютерная АКАДЕМИЯ

Программа курса «Разработка приложений с использованием WPF»

Розробка додатків з використанням WPF
Application Development using WPF

Цель курса

Обучить слушателя разработке приложений с использованием технологии WPF. Научить выбирать наиболее подходящий способ для отображения информации в приложении.

По окончании курса слушатель будет:

- уметь использовать XAML разметку;
- уметь проектировать каркасы пользовательских интерфейсов;
- владеть навыками по использованию стандартных элементов управления;
- владеть навыками по созданию собственных элементов управления;
- уметь связывать данные, из различных источников, с элементами пользовательского интерфейса;
- понимать принципы использования стилей;
- владеть навыками по работе с 2D-графикой;
- владеть навыками по отображению мультимедийной информации в приложении.

По окончании данного курса студент сдает практический экзамен по материалам курса. Для допуска к экзамену, должны быть сданы все домашние и практические задания.

Тематический план

Модуль 1. Паттерны MVP, MVVM

Модуль 2. Введение в WPF

Модуль 3. Контейнеры

Модуль 4. Элементы управления

Модуль 5. Отображение документов нефиксированного формата

Модуль 6. Соединение данных и элементов управления

Модуль 7. Управление стилями и ресурсами

Модуль 8. Создание пользовательских элементов управления

Модуль 9. Работа с графикой

Модуль 10. Экзамен

Модуль 1

Паттерны MVP, MVVM

1. Анализ паттерна MVC (Model-View-Controller).
2. Паттерн MVP:
   - что такое паттерн MVP?
   - цели и задачи паттерна MVP;
   - отличия паттернов MVC и MVP;
   - примеры использования паттерна MVP.
3. Паттерн MVVM:
   - что такое паттерн MVVM?
   - цели и задачи паттерна MVVM;
   - отличия паттернов MVC и MVVM;
   - примеры использования паттерна MVVM.

Модуль 2

Введение в WPF

1. История создания.
2. Новые возможности.
3. Архитектура WPF-приложения.
4. Обзор XAML:
   - понятие «Элемент»;
   - понятие «Атрибут»;
   - понятие «Связанные свойства»;
   - ознакомление с моделью Code Behind.
5. Обзор элементов:
   - контейнеры;
   - элементы управления;
   - графические фигуры.
6. Пример простой программы.

Модуль 3

Контейнеры

1. Grid.
2. DockPanel.
3. StackPanel.
4. Canvas.
5. ContentElement.
6. Практический пример сложного каркаса приложения с использованием разных контейнеров.

Модуль 4

Элементы управления

1. Button.
2. TextBox.
3. PasswordBox.
4. RadioButton.
5. Label.
6. Image.
7. Slider.
8. ProgressBar.
9. Menu.
10. ScrollBar.
11. ComboBox.
12. Expander.
13. ListBox.
14. TreeView.
15. ListView.
16. TabControl.
17. ToolBar.
18. ToolBarTray.
19. TextBlock.
20. Примеры использования.

Модуль 5

Отображение документов нефиксированного формата

1. Введение в документы нефиксированного формата.
2. Отображение документов:
   - элемент FlowDocumet;
   - элемент Paragraph;
   - элемент Figure.
3. Компоновка документа.
4. Примеры использования.

Модуль 6

Соединение данных и элементов управления

1. Общие принципы.
2. Рассмотрение класса ObjectDataSouce.
3. Рассмотрение класса XmlDataSource.
4. Рассмотрение связанных соединений.
5. Примеры использования.

Модуль 7

Управление стилями и ресурсами

1. Рассмотрение элементов для изменения внешнего вида приложения:
   - стили;
   - шаблоны;
   - скины;
   - темы.
2. Примеры использования.

Модуль 8

Создание пользовательских элементов управления

1. Рассмотрение создания пользовательского элемента управления.
2. Рассмотрение триггеров:
   - триггера свойств;

- триггера данных;
- триггера событий.

3. Использование пользовательских WPF-элементов управления в Windows Form-приложениях.
4. Примеры использования.

Модуль 9

Работа с графикой

1. Рассмотрение элементов:
   - rectangle;
   - ellipse;
   - line;
   - polyline;
   - polygon;
   - path;
   - brush;
   - pen.
2. Примеры использования.
3. Трансформации:
   - принципы работы трансформаций;
   - трансформации с помощью RenderTransform;
   - трансформация с помощью LayoutTransform;
   - классы для работы с трансформациями:
     - RotateTransform;
     - ScaleTransform;
     - SkewTransform;
     - MatrixTransform;
     - TranslateTransform;
     - TransformGroup.
   - примеры использования.
4. Отображение мультимедийной информации с помощью MediaElement.

Модуль 10

Экзамен

ШАГ компьютерная АКАДЕМИЯ

Программа курса
«Теория баз данных»

Теорія баз даних
Database Theory

Цель курса

Обучить слушателя основам теории баз данных. Объяснить принципы построения баз данных. Научить применять язык структурированных запросов SQL для взаимодействия с данными. Исследовать различные конструкции языка запросов SQL для правильного построения оптимальных запросов.

По окончании курса слушатель будет:

- уметь проектировать базы данных;
- применять нормальные формы для нормализации таблиц;
- уметь взаимодействовать с данными используя язык структурированных запросов SQL;
- создавать многотабличные запросы и подзапросы;
- использовать функции агрегирования;
- понимать принципы применения той или иной конструкции SQL.

По окончании данного курса студент сдает практическое задание 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.

Перед началом данного предмета необходимо предоставить студентам доступ к следующим курсам Microsoft Imagine Academy:

- Database Fundamentals.

Тематический план

Модуль 1. Введение в теорию баз данных

Модуль 2. Основы взаимодействия с MS SQL Server

Модуль 3. Запросы SELECT, INSERT, UPDATE, DELETE

Модуль 4. Многотабличные базы данных

Модуль 5. Функции агрегирования

Модуль 6. Объединения

Модуль 7. Экзамен

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;
   - сравнительный анализ СУБД Microsoft SQL Server с существующими системами управления базами данных.
2. Основы взаимодействия с Microsoft SQL Server:
   - версии и редакции Microsoft SQL Server;
   - инсталляция Microsoft SQL Server;
   - инструменты управления и утилиты MS SQL Server;
   - управление базой данных:
     - создание базы данных;
     - настройка параметров базы данных;
     - изменение размера базы данных;
     - переименование базы данных;
     - управление группами файлов;
     - удаление базы данных.

Модуль 2

Основы взаимодействия с MS SQL Server

1. Таблицы:
   - первичный ключ;
   - значение по умолчанию;
   - уникальность.

2. 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.
3. Индекс:
   - что такое индекс?
   - цели и задачи индекса;
   - внутреннее устройство индекса.
4. Системные базы данных и таблицы.
5. Запросы:
   - введение в язык структурированных запросов SQL;
   - язык SQL:
     - стандарты языка SQL;
     - диалекты языка SQL;
     - диалект Transact-SQL.
   - понятия DDL, DML, DCL.

Модуль 3

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.
6. Понятие транзакции. Использование транзакций.

Модуль 4

Многотабличные базы данных

1. Аномалии взаимодействия с однотабличной базой данных:
   - аномалии обновления;
   - аномалии вставки;
   - 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 5

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:
   - принципы использования HAVING;
   - сравнительный анализ HAVING и WHERE.

4. Подзапросы:
   - необходимость создания и использования подзапросов;
   - сравнение подзапросов и многотабличных запросов;
   - принцип работы подзапросов.

Модуль 6

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 7

Экзамен

ШАГ компьютерная АКАДЕМИЯ

Программа курса
«Программирование и администрирование СУБД MS SQL Server»

Програмування та адміністрування СКБД MS SQL Server
MS SQL Server Programming and Administration

Цель курса

Обучить слушателя основам программирования и администрирования MS SQL Server. Объяснить принципы программирования хранимых процедур, триггеров, пользовательских функций. Научить создавать представления. Исследовать механизмы безопасности MS SQL Server. Разобраться с аппаратом резервного копирования и репликации.

По окончании курса слушатель будет:

- уметь разрабатывать хранимые процедуры, триггеры, пользовательские функции;
- уметь создавать и изменять структуру базы данных с использованием механизмов DDL;
- разбираться в тонкостях использования индексов;
- применять механизмы резервного копирования/восстановления и репликации;
- понимать основы безопасности в MS SQL Server;
- уметь автоматизировать работу MS SQL Server.

По окончании данного курса студент сдает практическое задание 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. Работа с таблицами и представлениями в MS SQL Server

Модуль 2. Триггеры, хранимые процедуры и пользовательские функции

Модуль 3. Индексы

Модуль 4. Использование PIVOT, UNPIVOT. Обработка ошибок

Модуль 5. Безопасность

Модуль 6. Резервное копирование и восстановление

Модуль 7. Конфигурирование MS SQL Server. Activity Monitor. SQL Server Profiler

Модуль 8. Обмен данными между экземплярами MS SQL Server. Автоматизация задач

Модуль 9. Экзамен

Модуль 1

Работа с таблицами и представлениями в MS SQL Server

1. Создание таблиц. Оператор CREATE TABLE.
2. Модификация таблиц. Оператор ALTER TABLE.
3. Удаление таблиц. Оператор DROP TABLE.
4. Схема:
   - что такое схема?
   - цели и задачи схемы;
   - синтаксис создания схемы;
   - практические примеры использования.
5. Представления:
   - создание представлений;
   - модификация представлений;
   - удаление представлений;
   - изменения данных через представления.
6. In-memory таблицы.
7. Temporal таблицы.

Модуль 2

Триггеры, хранимые процедуры и пользовательские функции

1. Триггеры:
   - что такое триггер?
   - типы триггеров AFTER и INSTEAD OF;
   - триггер INSERT;
   - триггер UPDATE;
   - триггер DELETE.
2. Хранимые процедуры:
   - что такое хранимая процедура?
   - создание процедур;
   - выполнение процедур;
   - передача параметров;

- выходные параметры;
- возврат значений.

3. Пользовательские функции:
   - что такое пользовательская функция?
   - виды пользовательских функций:
     - скалярные функции (scalar function);
     - функции с табличными значениями (inline table function, multi-statement table function);
     - выполнение пользовательских функций;
     - отличие пользовательских функций и хранимых процедур.

Модуль 3

Индексы

1. Понятие индекса.
2. Виды индексов:
   - Clustered;
   - Nonclustered;
   - другие виды индексов:
     - Columnstrore;
     - Spatial;
     - XML;
     - Full-Text;
     - Index with included columns;
     - Filtered.
3. Создание, использование, удаление индексов.
4. Преимущества использования индексов.
5. Типичные проблемы при работе с индексами.
6. Execution plan:
   - что такое execution plan?
   - принципы формирования execution plan;
   - практические примеры использования.

Модуль 4

Использование PIVOT, UNPIVOT. Обработка ошибок

1. Что такое PIVOT?
2. Цели и задачи PIVOT.

3. Примеры использования PIVOT.
4. Что такое UNPIVOT?
5. Цели и задачи UNPIVOT.
6. Примеры использования UNPIVOT.
7. Обработка ошибок:
   - зачем обрабатывать ошибки?
   - использование TRY/CATCH;
   - использование THROW;
   - примеры обработки ошибок.

Модуль 5

Безопасность

1. Модель защиты данных.
2. Режимы защиты данных.
3. Пользователи базы данных.
4. Использование ролей. Роли уровня приложения.
5. Управление правами доступа:
   - специальные права доступа;
   - объектные права доступа;
   - командные права доступа;
   - цепочки подчинения.
6. Шифрование и безопасность. Инструмент Always Encrypted.
7. Инструмент data dynamic masking.

Модуль 6

Резервное копирование и восстановление

1. Резервное копирование. Необходимость и целесообразность.
2. Виды резервного копирования.
3. Средства резервного копирования Microsoft SQL Server.
4. Создание резервных копий.
5. Восстановление баз данных.
6. Виды восстановлений:
   - автоматическое восстановление;
   - ручное восстановление.

7. Восстановление резервных копий.
8. Восстановление баз данных по резервным копиям различных видов.
9. Восстановление поврежденных системных баз данных.

Модуль 7

Конфигурирование MS SQL Server. Activity Monitor. SQL Server Profiler

1. Конфигурирование SQL Server:
   - зачем конфигурировать SQL Server?
   - Microsoft SQL Server Configuration Manager;
   - конфигурирование с помощью Microsoft SQL Server Configuration Manager.
2. Activity Monitor.
3. Утилита SQL Profiler.
4. Анализ логов SQL Server.
5. Dynamic Management Views.
6. DBCC.

Модуль 8

Обмен данными между экземплярами MS SQL Server. Автоматизация задач

1. Обмен данными между экземплярами MS SQL Server:
   - обмен данными через резервное копирование и восстановление;
   - Copy Database Wizard;
   - bcp;
   - BULK INSERT.
2. Необходимость автоматизации задач:
   - автоматизация задач в Microsoft SQL Server;
   - утилита SQL Server Agent;
   - настройка SQL Server Agent;
   - практические примеры использования.

Модуль 9

Экзамен

Программа курса
«Технология доступа к базам данных ADO.NET»

Технологія доступу до баз даних ADO.NET
ADO.Net: Database Access Technology

Цель курса

Обучить слушателя разработке Windows-приложений с использованием платформы Microsoft.NET, языка программирования C# и технологии доступа к данным ADO.NET. Исследовать механизмы доступа к данным для использования в рамках Windows и Web приложений.

По окончании курса слушатель будет:

- уметь создавать Windows приложения с доступом к источникам данных;
- разбираться в технологиях доступа к данным;
- уметь выбирать правильный механизм доступа к источнику данных;
- уметь соединяться с базой данных, добавлять, удалять, обновлять данные;
- вызывать хранимые процедуры и передавать параметры;
- использовать механизм транзакций;
- уметь работать в присоединенном и отсоединенном режиме;
- уметь применять механизмы LINQ для работы с базами данных;
- использовать Entity Framework для взаимодействия с источниками данных;
- знать особенности работы с Dapper.

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

Перед началом данного предмета необходимо предоставить студентам доступ к следующим курсам Microsoft Imagine Academy:

- Using Data in Software Applications.

Тематический план

Модуль 1. Введение в ADO.NET

Модуль 2. Присоединенный режим

Модуль 3. Фабрика провайдеров, асинхронный режим доступа, конфигурационные файлы

Модуль 4. Отсоединенный режим

Модуль 5. LINQ to SQL

Модуль 6. Введение в Entity Framework

Модуль 7. Использование Dapper

Модуль 1

Введение в ADO.NET

1. Что такое ADO.NET?
2. Исторический экскурс в технологии доступа к данным:
   - ODBC;
   - DAO;
   - OLEDB;
   - ADO.
3. Сравнительный анализ технологий доступа к данным.
4. Сравнительный анализ понятий драйвер и провайдер.
5. Пространства ADO.NET:
   - System.Data;
   - System.Data.Common;
   - System.Data.OleDb;
   - System.Data.SqlClient;
   - System.Data.Sql;
   - System.Data.Odbc;
   - System.Data.OracleClient;
   - другие пространства.
6. Модели работы ADO.NET:
   - присоединенный режим;
   - отсоединенный режим.
7. Концепция интерфейсов и базовых классов ADO.NET:
   - интерфейс IDbConnection;
   - интерфейс IDbCommand;
   - интерфейсы IDataReader, IDataRecord;
   - интерфейсы IDataAdapter, IDbDataAdapter;
   - интерфейсы IDataParameter, IDbDataParameter;
   - интерфейс IDbTransaction;
   - классы DbConnection, DbCommand, DbDataReader, DbDataAdapter, DbParameter, DbTransaction.
8. Обзорный пример использования ADO.NET для доступа к источнику данных.

Модуль 2

Присоединенный режим

1. Класс DbConnection:
   - цели и задачи класса DbConnection и его потомков;
   - анализ методов и свойств;
   - пример соединения с источником данных.
2. Класс DbCommand:
   - цели и задачи класса DbCommand и его потомков;
   - анализ методов и свойств;
   - пример посылки запроса.
3. Класс DbDataReader:
   - цели и задачи класса DbDataReader и его потомков;
   - анализ методов и свойств;
   - пример получения данных.
4. Примеры вставки, обновления, удаления данных.
5. Использование параметров:
   - класс DbParameter:
     - цели и задачи класса DbParameter и его потомков;
     - анализ методов и свойств;
     - пример передачи параметров.
   - использование и вызов хранимых процедур.
6. Использование транзакций:
   - класс DbTransaction:
     - цели и задачи класса DbTransaction и его потомков;
     - анализ методов и свойств;
     - пример работы с транзакциями.
   - использование механизма транзакций аппарата СУБД.

Модуль 3

Фабрика провайдеров, асинхронный режим доступа, конфигурационные файлы

1. Фабрика провайдеров ADO.NET:
   - концепция фабрики провайдеров;
   - класс DbProviderFactories:
     - цели и задачи класса DbProviderFactories и его потомков;

- анализ методов класса;
- пример использования получения конкретной фабрики.

- фабрики, специфичные для провайдера:
  - цели и задачи класса DbProviderFactory и его потомков;
  - анализ методов и свойств;
  - пример использования конкретной фабрики.

2. Асинхронные механизмы доступа к данным:
   - что такое асинхронность?
   - зачем нужна асинхронность?
   - анализ методов для асинхронного механизма доступа к данным;
   - примеры использования асинхронного доступа.
3. Использование конфигурационных файлов:
   - что такое конфигурационный файл?
   - цели и задачи конфигурационных файлов;
   - типы конфигурационных файлов;
   - способы доступа к конфигурационным файлам;
   - примеры использования конфигурационных файлов при доступе к источникам данных.

Модуль 4

Отсоединенный режим

1. Что такое отсоединенный режим?
2. Концепция использования отсоединенного режима.
3. Что такое DataSet?
   - цели и задачи класса DataSet;
   - анализ методов и свойств;
   - пример использования DataSet.
4. Класс DataTable:
   - цели и задачи класса DataTable;
   - анализ методов и свойств;
   - пример использования DataTable.
5. Класс DataRow:
   - цели и задачи класса DataRow;
   - анализ методов и свойств;
   - пример использования DataRow.

6. Класс DataColumn:
   - цели и задачи класса DataColumn;
   - анализ методов и свойств;
   - пример использования DataColumn.
7. Класс DbDataAdapter:
   - цели и задачи класса DbDataAdapter;
   - анализ методов и свойств;
   - пример использования DbDataAdapter.
8. Класс SqlCommandBuilder:
   - Цели и задачи класса SqlCommandBuilder;
   - Анализ методов и свойств;
   - Пример использования SqlCommandBuilder.

Модуль 5

LINQ to SQL

1. Что такое LINQ to SQL?
2. Цели и задачи LINQ to SQL.
3. Схема работы LINQ to SQL.
4. Фильтрация данных.
5. Сортировка данных.
6. Группировка данных.
7. Вложенные запросы.
8. Объединения (join).
9. Обновление данных.
10. Вставка данных.
11. Удаление данных.

Модуль 6

Введение в Entity Framework

1. Что такое Entity Framework?
2. Цели и задачи Entity Framework.
3. Понятие модели.
4. Понятие сгенерированного кода.

5. Обзор различных подходов при работе с EF:
   - DB First;
   - Model First;
   - Code First.
6. Использование DB First:
   - почему стоит использовать DB First?
   - создание модели;
   - изменение модели.
7. Использование Model First:
   - почему стоит использовать Model First?
   - создание модели;
   - изменение модели.
8. Использование Code First:
   - почему стоит использовать Code First?
   - создание модели;
   - понятие domain classes;
   - изменение модели.
9. Примеры использования.

Модуль 7

Использование Dapper

1. Что такое Dapper?
2. Цели и задачи Dapper.
3. Сравнение Dapper и Entity Framework.
4. Инсталляция и настройка Dapper.
5. Соединение с базой данных.
6. Получение данных из базы данных.
7. Вставка данных.
8. Обновление данных.
9. Удаление данных.
10. Исполнение хранимых процедур.

Программа курса
«Системное программирование»

Системне програмування
System programming

Цель курса

Обучить слушателя разработке Windows-приложений с использованием механизмов системного программирования. Получить теоретические и практические знания об управлении памятью в Windows. Изучить организацию динамических модулей в Windows. Разработать приложения, использующие системные ловушки (hooks), многопоточность. Выяснить принципы синхронизации потоков в Windows. Выяснить принципы взаимодействия с системной базой данных Windows (реестр).

По окончании курса слушатель будет:

- взаимодействовать с унаследованным кодом;
- разбираться в основах архитектуры Microsoft Windows;
- использовать механизмы синхронизации;
- порождать процессы;
- создавать многопоточные приложения;
- уметь устанавливать системные ловушки;
- разрабатывать динамически подключаемые библиотеки.

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.

Важно отметить, что в данном курсе вопросы системного программирования рассматриваются в разрезе Win API и .NET Framework.

Тематический план

Модуль 1. Использование унаследованного программного кода

Модуль 2. Процессы

Модуль 3. Многопоточность и асинхронность

Модуль 4. Синхронизация

Модуль 5. Параллельное программирование

Модуль 6. Управление памятью

Модуль 7. Использование реестра

Модуль 8. Динамически подключаемые библиотеки

Модуль 9. Хуки

Модуль 10. Экзамен

Модуль 1

Использование унаследованного программного кода

1. Почему необходимо использовать унаследованный программный код.
2. Пространство System.Runtime.InteropServices.
3. Взаимодействие с модулями Dll:
   - класс DllImportAttribute;
   - поле ExactSpelling;
   - поле EntryPoint;
   - поле CharSet;
   - поле CallingConvention;
   - поле SetLastError.
4. Примеры использования.

Модуль 2

Процессы

1. Основные сведения о процессах.
2. Функции манипулирования процессами.
3. Понятие дочернего процесса.
4. Манипулирование дочерним процессом.
5. Примеры создания процессов в Win API и .NET Framework.
6. Домен приложения.
7. Использование доменов приложения.

Модуль 3

Многопоточность и асинхронность

1. Что такое многопоточность?
2. Общие сведения о потоках:
   - что такое поток?
   - порождение потоков;

- приостановка, возобновление, прекращение потока;
- приоритеты потоков.

3. Потоки в .NET Framework:
   - пространство System.Threading;
   - класс Thread;
   - потоки фоновые и первичные;
   - порождение потоков;
   - приостановка, возобновление, прекращение потока;
   - приоритеты потоков.
4. Потоки в Win API:
   - создание потока;
   - потоковая функція;
   - остановка потока;
   - передача параметров потоковой функции;
   - приостановка, возобновление, прекращение потока;
   - приоритеты потоков.
5. Асинхронный вызов методов.
6. Использование таймеров обратного вызова.
7. Использование пула потоков.

Модуль 4

Синхронизация

1. Проблемы синхронизации.
2. Объекты синхронизации:
   - мьютексы;
   - семафоры;
   - события;
   - критические секции;
   - взаимоисключающий доступ.
3. Мьютексы в .NET Framework. Класс Mutex.
4. Семафоры в .NET Framework. Класс Semaphore.
5. События в .NET Framework.
   - класс ManualResetEvent;
   - класс AutoResetEvent.

6. Критическая секция в .NET Framework:
   - класс Monitor;
   - ключевое слово lock.
7. Использование Volatile.
8. Взаимоисключающий доступ в .NET Framework. Класс Interlocked.
9. Использование объектов синхронизации в Win API:
   - мьютексы;
   - семафоры;
   - события;
   - критические секции.

Модуль 5

Параллельное программирование

1. Что такое параллельное программирование?
2. Цели и задачи параллельного программирования.
3. Что такое Task Parallel Library?
4. Класс Task:
   - цели и задачи класса Task;
   - методы класса Task;
   - свойства класса Task;
   - примеры использования класса Task.
5. Класс Parallel:
   - цели и задачи класса Parallel;
   - методы класса Parallel;
   - примеры использования класса Parallel.
6. Структуры данных для параллельного программирования.
7. PLINQ:
   - что такое Parallel LINQ;
   - отличия LINQ от PLINQ;
   - класс ParallelEnumerable:
     - цели и задачи ParallelEnumerable;
     - методы класса ParallelEnumerable;
     - примеры использования класса ParallelEnumerable.

- скорость выполнения в PLINQ:
  - факторы, влияющие на скорость выполнения;
  - практические примеры.
- сохранение порядка;
- параметры для слияния.

Модуль 6

Управление памятью

1. Управление памятью в Windows.
2. Структура виртуального адресного пространства процесса.
3. Понятие региона памяти, страницы памяти, страничного файла.
4. Трансляция виртуального адреса на физический.
5. Защита памяти.
6. Особенности управления памятью при разработке приложений платформы Microsoft .NET.
7. Стек потока.
8. Куча процесса по умолчанию (стандартная куча).

Модуль 7

Использование реестра

1. Основные сведения о реестре.
2. Работа с реестром с помощью Win API.
3. Работа с реестром с помощью платформы Microsoft.NET.

Модуль 8

Динамически подключаемые библиотеки

1. Принципы создания динамически подключаемых библиотек в Visual C++.
2. Неявное связывание.
3. Явное связывание.
4. Отложенная загрузка.
5. Анализ функции DllMain.
6. Разработка динамически подключаемых библиотек с использованием .NET Framework.

Модуль 9

Хуки

1. Создание и использование ловушек.
2. Функции Win API для работы с хуками.
3. Создание хуков в приложениях платформы Microsoft .NET.

Модуль 10

Экзамен

Программа курса
«Сетевое программирование»

Мережеве програмування
Network programming

Цель курса

Обучить слушателя разработке Windows-приложений с использованием механизмов сетевого взаимодействия. Получить теоретические и практические знания об основах сетевого программирования в Windows.

По окончании курса слушатель будет:

- уметь строить клиент-серверные приложения с использованием сокетов;
- использовать механизмы асинхронности при работе с сокетами;
- осуществлять широковещательную рассылку;
- посылать почту с использованием пространства System.Net.Mail;
- получать и посылать данные по FTP протоколу;
- использовать протокол HTTP для получения и отправки данных.

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 сети

Модуль 2. Сокеты

Модуль 3. TCP и UDP сокеты

Модуль 4. Unicast, Broadcast, Multicast

Модуль 5. HTTP, SMTP, FTP

Модуль 6. Экзамен

Модуль 1

Введение в сети

1. Что такое сетевое программирование?
2. Цели и задачи сетевого программирования.
3. Что такое сеть?
4. Типы сетей.
5. Модель OSI.
6. Базовые термины:
   - сетевые протоколы;
   - IP адрес;
   - сокет;
   - порт.

Модуль 2

Сокеты

1. Типы сокетов.
2. Класс Socket:
   - общая последовательность вызовов для серверной и клиентской части;
   - цели и задачи класса Socket;
   - анализ методов необходимых для построения серверной части:
     - метод Bind;
     - метод Listen;
     - метод Accept;
     - методы Receive/Send;
     - метод Close.
   - анализ методов необходимых для построения клиентской части:
     - метод Connect;
     - методы Receive/Send;
     - метод Close.
3. Пример построения клиент/серверного приложения с использованием сокетов.

4. Асинхронные сокеты:
   - цели и задачи асинхронных сокетов;
   - общая последовательность вызовов для серверной и клиентской части;
   - анализ методов необходимых для построения серверной части:
     - метод BeginAccept;
     - метод BeginReceive;
     - метод BeginSend;
     - метод EndAccept;
     - метод EndReceive;
     - метод EndSend.
   - анализ методов необходимых для построения клиентской части:
     - метод BeginConnect;
     - метод BeginReceive;
     - метод BeginSend;
     - метод EndConnect;
     - метод EndReceive;
     - метод EndSend.
5. Пример построения клиент/серверного приложения с использованием асинхронных сокетов.

Модуль 3

TCP и UDP сокеты

1. Обзор TCP протокола:
   - общий обзор;
   - терминология TCP;
   - TCP заголовки;
   - преимущества и недостатки TCP.
2. Обзор UDP протокола:
   - общий обзор;
   - терминология UDP;
   - преимущества и недостатки UDP.
3. Использование классов, характерных для TCP:
   - класс TcpListener;
   - класс TcpClient;
   - пример использования.

4. Использование классов, характерных для UDP:
   - класс UdpClient;
   - пример использования.
5. Использование сокетов в программе на C++:
   - функция socket;
   - функция setsockopt;
   - функция bind;
   - функция listen;
   - функция accept;
   - функция connect;
   - практические примеры использования.

Модуль 4

Unicast, Broadcast, Multicast

1. Что такое Unicast?
2. Что такое Broadcast?
3. Что такое Multicast?
4. Пример реализация Multicast приложения.

Модуль 5

HTTP, SMTP, FTP

1. Обзор HTTP протокола:
   - общий обзор;
   - терминология HTTP;
   - HTTP заголовки;
   - отправка запросов и получение ответов.
2. Классы для работы с HTTP:
   - отправка запросов с использованием класса HttpWebRequest;
   - получение ответов с использованием класса HttpWebResponse;
   - использование класса WebClient;
   - примеры использования.
3. Работа с электронной почтой:
   - общий обзор почтовых протоколов SMTP, POP3, IMAP;

- SMTP:
  - общий обзор;
  - терминология SMTP;
  - команды SMTP;
  - коды ответов;
  - анализ формата электронного письма;
  - что такое MIME?
- POP3:
  - общий обзор;
  - терминология POP3;
  - команды POP;
  - коды ответов.
- IMAP:
  - общий обзор.
- классы для работы SMTP:
  - пространство System.Net.Mail;
  - класс MailMessage;
  - класс Attachment;
  - класс SmtpClient;
  - пример отправки почты.

4. Использование FTP:
   - общий обзор;
   - терминология FTP;
   - пример типичной FTP сессии;
   - классы для работы с FTP:
     - класс FtpWebRequest;
     - класс FtpWebResponse.
   - пример использования FTP.

Модуль 6

Экзамен

Программа курса
«Управление программными проектами»

Керування програмними проектами
Software Project Management

Цель курса

Ввести слушателя в предметную область «Управления программными проектами». Научить разбираться и использовать различные модели процесса разработки при построении программного обеспечения.

По окончании курса слушатель будет:

- уметь строить диаграммы Ганта;
- понимать отличия командной и одиночной разработки;
- уметь оперировать терминами предметной области;
- разбираться в отличиях моделей разработки ПО;
- высчитывать риски, присущие проекту;
- использовать инструментальные средства для обеспечения цикла разработки проекта;
- уметь создавать и анализировать документацию проекта.

В конце данного курса студент выполняет практическое задание на основании, которого выставляется оценка по курсу.

Тематический план

Модуль 1. Введение в управление программными проектами

Модуль 2. Детальней об управлении проектом

Модуль 3. Работа с требованиями

Модуль 4. Подробнее о Scrum

Модуль 5. Практическое задание, исполняемое в рамках Team Foundation Server

Модуль 1

Введение в управление программными проектами

1. Введение в предметную область.
2. Причины возникновения дисциплины «управление программными проектами».
3. Диаграммы Ганта.
4. Что такое проект и программный проект?
5. Что такое жизненный цикл процесса разработки программного обеспечения?
6. Что такое управление проектами?
7. Что такое одиночная разработка?
8. Что такое командная разработка?
9. Анализ проблем одиночной и командной разработки программного обеспечения.
10. Анализ терминов предметной области:
    - процесс;
    - проект;
    - персонал;
    - продукт;
    - качество.
11. Характеристики проекта:
    - тип проекта;
    - цель проекта;
    - требования к качеству;
    - требования к бюджету;
    - требования по срокам завершения.
12. Расходы, связанные с проектом:
    - прямые;
    - непрямые.
13. 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.

14. Управление качеством.
15. Документирование.

Модуль 2

Детальней об управлении проектом

1. Проект:
   - составляющие управления проектом;
   - параметры проекта:
     - стоимость:
       - что такое стоимость проекта;
       - характеристики, влияющие на стоимость проекта;
       - принципы оценки стоимости;
       - примеры расчетов оценки стоимости.
     - функциональность;
     - качество;
     - расписание.
   - участники и персонал проекта:
     - участники проекта со стороны заказчика;
     - персонал проекта со стороны фирмы разработчика:
       - принципы отбора персонала;
       - управление персоналом.

- роли в рамках проекта:
  - разработчик;
  - тестер;
  - бизнес-аналитик;
  - Project manager;
  - архитектор;
  - Team Leader;
  - другие роли.

2. Риски в проекте:
   - что такое риски?
   - типы рисков;
   - принципы управления рисками;
   - выявление рисков;
   - предупреждение рисков.
3. Управление качеством в проекте:
   - что такое управление качеством?
   - метрики;
   - план контроля качества;
   - практическое использование метрик.
4. Документация и документооборот:
   - цели и задачи документации в рамках проекта;
   - типы документации;
   - документы, необходимые в рамках каждой фазы процесса:
     - Vision & Scope;
     - Project requirements;
     - Design specification;
     - Test plan;
     - другие документы.

Модуль 3

Работа с требованиями

1. Что такое требование?
2. Что такое анализ требований?
3. C-Требования (требования заказчика) и D-требования (требования разработчика).

4. Типичная схема анализа требований.
5. Преимущества и недостатки анализа требований.
6. Методологии для выработки C-требований.
7. Типы D-требований.
8. Свойства D-требований.
9. Методологии для выработки D-требований.

Модуль 4

Подробнее о Scrum

1. Что такое Scrum?
2. Причины возникновения Scrum.
3. Роли в Scrum:
   - владелец продукта;
   - команда;
   - Scrum-мастер.
4. Бэклог продукта:
   - что такое бэклог продукта?
   - как создавать бэклог?
   - как оценивать задачи в бэклоге?
   - что такое scrum-доска?
   - примеры создания бэклога.
5. Спринт:
   - что такое спринт?
   - планирование спринтов;
   - ежедневный скрам;
   - обзор спринта;
   - ретроспективное собрание.
6. 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-v2.0.pdf.
7. Утилиты и инструментальные средства, используемые при работе в проектах:
   - системы контроля версий:
     - SVN;

  - Git;
  - CVS.
- баг-трекеры:
  - Bugzilla;
  - Mantis.
- другие инструменты.

Модуль 5

Практическое задание, исполняемое в рамках Team Foundation Server

ШАГ компьютерная АКАДЕМИЯ

Программа курса
«Разработка веб-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3»

Розробка веб-сторінок на мові розмітки HTML5
з використанням каскадних таблиць стилів CSS3

Creating Web-pages Using HTML5 and CSS3

Цель курса

Обучить слушателя созданию и верстке статических web-страниц с использованием технологий XHTML1.0, HTML4/5, CSS2/3. 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 практический экзамен по материалам курса. Для допуска к экзамену, должны быть сданы все домашние и практические задания.

Перед началом данного предмета необходимо предоставить студентам доступ к следующим курсам Microsoft Imagine Academy:

- Developing in HTML5 with JavaScript and CSS3 Jump Start;
- Preparing for Exam MTA 98-375 HTML5 App Development Fundamentals: Academic Edition.

Тематический план

Модуль 1. Введение в Web-технологии. Структура HTML

Модуль 2. Форматирование текста при помощи HTML

Модуль 3. Форматирование при помощи CSS

Модуль 4. Списки. CSS отступы и поля

Модуль 5. Графика в web-дизайне. Оптимизация графики

Модуль 6. Гиперссылки. Принципы навигации web-сайта

Модуль 7. Таблицы

Модуль 8. Позиционирование. Верстка web-страниц блоками

Модуль 9. Формы. Фреймы

Модуль 10. Первичная оптимизация сайта. Размещение сайта в сети Internet

Модуль 11. Создание страниц посредством HTML5 и CSS3

Модуль 12. Экзамен

Модуль 1

Введение в Web-технологии. Структура HTML

1. Введение в предмет.
2. Введение в языки разметки. Язык разметки гипертекста HTML:
   - развитие HTML, версии. Текущие используемые версии: HTML и XHTML;
   - вопросы межбраузерной совместимости. Война браузеров;
   - W3C.
3. Теги – основной элемент структуры HTML. Правила записи тегов и их атрибутов в стандарте XHTML на примере тегов <b>, <i>, <u>, <font>, <sup>, <sub>,
. Синтаксические отличия HTML4 от XHTML.
4. Основные ошибки в записях тегов:
   - спецификации <!DOCTYPE HTML>;
   - валидация документа при помощи FireFox – дополнение HTML Validator;
   - понятие well-formed;
   - прародители HTML4/5 и XHTML: SGML и XML.
5. Основная структура XHTML документа. Основные элементы и их назначение.
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.

Модуль 2

Форматирование текста при помощи HTML

1. Классификация тегов: линейные и блочные:
   - линейные: <b>, <i>, <u>, <font>;
   - блочные: <p>, <h1>.<h6>.
2. Модель форматирования текста: заголовки и абзацы. Элементы <p>, <h1>.<h6>. Выравнивание текста в блочных элементах: атрибут align.

3. 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>.
4. Цвета в Web:
   - Web-палитра;
   - Photoshop/GIMP – дополнительные инструменты верстальщика. Рассмотрение основных возможностей: открытие изображений, подбор цветов при помощи палитры Photoshop|GIMP, инструмент пипетка;
   - использование выбранного цвета в атрибуте color тега <font>.
5. Практика: создание простейшей web-страницы.

Модуль 3

Форматирование при помощи CSS

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.
6. Практика: форматирование текста при помощи CSS.

Модуль 4

Списки. CSS отступы и поля

1. Создание списков:
   - неупорядоченные списки: элементы <ul>, <li>;
   - упорядоченные списки: элементы <ol>, <li>;
   - атрибуты type, value, start.
2. Создание вложенных списков.
3. 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.
4. Списки определений: элементы <dl>, <dd>, <dt>.
5. 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>.
6. Практика: создание списков.

Модуль 5

Графика в web-дизайне. Оптимизация графики

1. Форматы графических файлов в Web.
2. Рассмотрение инструментов Photoshop/GIMP для работы с изображениями:
   - слои в Photoshop/GIMP;
   - прямоугольное выделение. Линейки. Направляющие;
   - оптимизация изображений в Photoshop.
3. 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.
4. 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.
5. Спрайты: меньше картинок – больше скорость. Создание спрайтов при помощи онлайн сервисов (spritegen.website-performance.org, csssprites.com, printf.ru/spritr).
6. Практика: проектирование web-страниц с использованием графики.

Модуль 6

Гиперссылки. Принципы навигации web-сайта

1. Общие сведения о гиперссылках:
   - тег <a> и его атрибуты (href, target);
   - эргономика, удобство навигации.
2. 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.

3. Создание меню при помощи структуры списков (<ul>, <li>), его форматирование. Свойство display. Преобразование ссылки в блочный элемент.
4. Псевдоклассы:
   - псевдоклассы ссылок: active, hover, link, visited;
   - псевдоклассы для обычных элементов: first-child, first-line, first-letter.
5. CSS свойство cursor.
6. Практика: работа по разработке галереи изображений.

Модуль 7

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.
5. Практика: создание сложных таблиц.
6. Основы табличной верстки. Пример табличной верстки: ее минусы.

Модуль 8

Позиционирование. Верстка web-страниц блоками

1. Свойство position:
   - рассмотрение позиционирования: relative и absolute;
   - свойства top, left, bottom, right.
2. Свойства visibility, overflow.
3. Практика.
4. 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.
5. Рассмотрение простейших структур страниц:
   - структура фиксированного размера.
6. Резиновая структура. Блоки с отрицательными margin.

Модуль 9

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. Форматирование элементов форм при помощи CSS.
5. Фреймы и их структура (теоретические сведения).
   - тег <iframe>;
   - использование Спецификации <!DOCTYPE HTML> для фреймов;
   - вредность использования фреймов;
   - применение тега <iframe> в визуальных редакторах WYSIWYG.

Модуль 10

Первичная оптимизация сайта. Размещение сайта в сети Internet

1. Оптимизация сайта под поисковые системы.
   - Задание ключевых слов и описания при помощи тега <meta>;
   - важность использования заголовков <h1>.<h6> и логических тегов и ;
   - важность использования уникального содержимого.
2. Хостинг, регистрация доменного имени.
3. Размещение сайта в Internet. Работа с FTP-клиентами.
4. Регистрация в поисковых системах и каталогах. Размещение счетчика на web-странице.

Модуль 11

Создание страниц посредством HTML5 и CSS3

1. Структура HTML5 документа:
   - новые теги задания структуры: <header>, <nav>, <section>, <article>, <aside>, <footer>. Доступность новых тегов в современных браузерах. Отображение новых тегов в устаревших браузерах.

2. HTML5 – конкурент Flash:
   - вставка видео на странице посредством тега <video>;
   - вставка аудио на странице посредством тега <audio>;
   - создание изображений и анимации посредством тега <canvas>;
   - использование SVG формата.
3. Новые элементы форм.
4. Новые свойства: 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.

Модуль 12

Экзамен

Создание web-сайта с последующим размещением в Internet.
Основные требования: блочная верстка, валидный код.

ШАГ компьютерная АКАДЕМИЯ

Программа курса «Язык сценариев JavaScript и библиотека jQuery»

Мова сценаріїв JavaScript та бібліотека JQuery
JavaScript Programming Language and jQuery

Цель курса

Обучить студента разработке клиентских сценариев с использованием JavaScript. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи. Обучить студента особенностям использования библиотеки jQuery при разработке клиентских сценариев. Научить выбирать правильные механизмы и конструкции библиотеки jQuery 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;
- владеть базовыми конструкциями библиотеки jQuery;
- разбираться в тонкостях использования того или иного селектора;
- уметь создавать обработчики событий и воздействовать на поведение событий;
- уметь взаимодействовать и изменять стили веб-страницы;
- владеть способами внедрения анимации с использованием jQuery;
- обладать знаниями для воздействия на структуру документа;
- применять механизмы jQuery для работы с Ajax;
- уметь подключать и использовать jQuery плагины.

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 JavaScript

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

Модуль 3. Обработка событий

Модуль 4. Browser Object Model. Document Object Model

Модуль 5. Формы

Модуль 6. Проверка достоверности форм. Использование Cookie

Модуль 7. Рисование с помощью canvas

Модуль 8. HTML5 и JavaScript

Модуль 9. JSON

Модуль 10. Ajax

Модуль 11. ECMAScript

Модуль 12. Введение в jQuery

Модуль 13. События и jQuery

Модуль 14. Стили и анимация

Модуль 15. Взаимодействие с DOM

Модуль 16. AJAX и jQuery

Модуль 17. Использование jQuery плагинов

Модуль 18. Экзамен

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-аксессоры:
    - 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

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.

Модуль 8

HTML5 и JavaScript

1. Cross-document messaging или XDM:
   - цели и задачи XDM;
   - отправка сообщений. Метод postMessage;
   - получение сообщений.

2. Drag and Drop:
   - поддержка drag and drop в различных браузерах;
   - события, возникающие при drag and drop;
   - объект dataTransfer:
     - методы объекта dataTransfer;
     - свойства dropEffect и effectAllowed.
   - свойство draggable.
3. Поддержка медиавозможностей:
   - использование тега <video>;
   - использование тега <audio>.

Модуль 9

JSON

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.

Модуль 10

Ajax

1. Синхронные и асинхронные запросы.
2. Что такое Ajax?
3. Объект XMLHttpRequest:
   - создание через ActiveX объект;
   - создание через объект XMLHttpRequest.

4. Методы и свойства XMLHttpRequest.
5. Понятие HTTP заголовка.
6. Использование метода GET. URL кодирование.
7. Использование метода POST.

Модуль 11

ECMAScript

1. Что такое ECMAScript?
2. История возникновения.
3. Версии ECMAScript.
4. ECMAScript 6, 7, 8.
5. Переменные.
6. Тип данных Symbol.
7. Функции-стрелки.
8. Использование строк.
9. Объекты и классы.
10. Модули.
11. Другие возможности ECMAScript 6.
12. Возможности ECMAScript 7.
13. Возможности ECMAScript 8.

Модуль 12

Введение в jQuery

1. Что такое jQuery?
2. Цели и задачи jQuery.
3. История создания jQuery.
4. Версии jQuery.
5. Подключение jQuery.
6. Доступ к элементам страницы при помощи функции $.
7. Понятие селектора.

8. Типы селекторов:
   - CSS селекторы;
   - jQuery селекторы.
9. Traversing. Методы обхода DOM. Метод filter, next, nextAll, prev, prevAll, siblings и др.

Модуль 13

События и jQuery

1. Создание обработчиков событий с использованием jQuery.
2. Удаление обработчиков событий.
3. Объект Event и jQuery.
4. Воздействие на обработку события.
5. Запуск обработки события.

Модуль 14

Стили и анимация

1. Метод css.
2. Отображение и скрытие элементов. Методы show и hide.
3. Создание эффектов.
4. Анимация.

Модуль 15

Взаимодействие с DOM

1. Создание новых элементов DOM.
2. Вставка элементов DOM.
3. Передвижение элементов DOM.
4. Копирование элементов DOM.
5. Взаимодействие с атрибутами.

Модуль 16

AJAX и jQuery

1. JSON.
2. Механизмы Ajax внутри библиотеки jQuery.
3. Использование метода GET.
4. Использование метода POST.
5. События и Ajax в рамках jQuery.
6. Обработка ошибок.

Модуль 17

Использование jQuery плагинов

1. Понятие плагина jQuery.
2. Подключение плагина.
3. Примеры плагинов:
   - Cycle;
   - jQuery UI.

Модуль 18

Экзамен

Программа курса
«Разработка веб-приложений с использованием технологий ASP.NET и AJAX»

Розробка веб-додатків з використанням технологій ASP.NET і AJAX
Web Application Development Using ASP.Net and AJAX

Цель курса

Обучить студента разработке серверных решений с использованием ASP.NET Web Forms и ASP.NET MVC. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

По окончании курса слушатель будет:

- разбираться в принципах построения ASP.NET приложений;
- понимать архитектуру Model-View-Controller;
- использовать элементы управления;
- проверять данные, введенные пользователем;
- взаимодействовать с источниками данных;
- применять AJAX для построения веб-приложений;
- обеспечивать безопасность работы ASP.NET приложений;
- создавать ASP.NET MVC приложения;
- уметь настраивать маршруты;
- использовать фильтры;
- применять селекторы;
- использовать методы HtmlHelper для создания элементов управления;
- использовать возможности движка Razor;
- применять css-фреймворк Bootstrap для ASP.NET MVC приложений;
- создавать веб-сервисы с помощью Web API.

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.

Перед началом данного предмета необходимо предоставить студентам доступ к следующим курсам Microsoft Imagine Academy:

- Course 70001: Web Application Architecture and Design;
- Developing ASP.NET MVC 4 Web Applications Jump Start;
- Course 70004: Server-Side Code for Web Forms;
- Course 70008: Advanced Ajax, ASP.NET Deployments, and Silverlight.

Тематический план

Модуль 1. Введение в ASP.NET. Знакомство с Web Forms

Модуль 2. Inversion of Control. Dependency Injection

Модуль 3. Введение в ASP.NET MVC

Модуль 4. Контроллеры. Класс ActionResult

Модуль 5. Представления. Модели. Класс HtmlHelper. Движок Razor

Модуль 6. Использование источников данных в ASP.NET MVC приложениях

Модуль 7. Использование AJAX в ASP.NET MVC приложениях, безопасность ASP.NET MVC приложений

Модуль 8. Использование Bootstrap в ASP.NET MVC приложениях

Модуль 9. Web API

Модуль 10. ASP.NET Core

Модуль 11. Экзамен

Модуль 1

Введение в ASP.NET. Знакомство с Web Forms

1. История развития серверных технологий от компании Microsoft.
   - ASP.
   - ASP.NET:
     - ASP.NET Web Forms;
     - ASP.NET MVC;
     - ASP.NET Core.
2. Архитектура WEB приложения.
3. Виды WEB приложений.
   - Страничные.
   - Компонентные.
   - Модель – Вид – Контролер.
4. Архитектура ASP.NET приложения.
   - Классы HttpContext, HttpApplication, HttpSession, HttpRequest, HttpResponse, HttpCookie, HttpViewState, HttpCache.
   - Модель Code-Behind.
   - Конфигурирование приложения. Файлы конфигурации (machine.config, web.config).
   - Файл Global.asax.
   - Жизненный цикл страницы (класс Page).
   - Директивы в ASP.NET.
5. Работа с HTML формами в ASP.NET.
6. Атрибут runat="server".
7. Серверные элементы управления.
8. События в ASP.NET.
9. Свойство Page.IsPostBack.
10. Обзор серверных элементов управления.
    - Простые элементы управления:
      - Button;
      - TextEdit;
      - Label;

- Image;
- Suggestion;
- CheckBox;
- RadioButton;
- UploadImage.

- Списочные элементы управления:
  - ListBox;
  - DropDown;
  - CheckListBox;
  - Repeater.

11. Практические примеры использования.

Модуль 2

Inversion of Control. Dependency Injection

1. Inversion of Control.
   - Что такое Inversion of Control?
   - Цели и задачи Inversion of Control.
   - Практические примеры.
2. Dependency Injection.
   - Что такое Dependency Injection?
   - Цели и задачи Dependency Injection.
   - Практические примеры.
3. DI фреймворки.
   - Autofac.
   - Ninject.
   - Практические примеры использования.

Модуль 3

Введение в ASP.NET MVC

1. Архитектура Модель – Вид – Контроллер.
   - Принципы архитектуры MVC.
   - Модель.
   - Вид.

- Контроллер.
- Сравнение MVC с компонентным подходом.
- Готовые шаблоны MVC-приложений.

2. Создание базового приложения ASP.NET MVC в Visual Studio.
   - Создание пустого проекта ASP.NET MVC.
   - Добавление контроллера.
   - Запуск проекта.
3. Жизненный цикл работы приложения.
4. Жизненный цикл обработки запроса.
5. Маршруты (routes).
   - Что такое маршрут?
   - Что такое роутинг (маршрутизация)?
   - Принципы работы маршрутов.
   - Связь маршрута и контроллера.
   - Пространство System.Web.Routing.
   - Класс RouteData.
   - Класс RouteCollection.
   - Практический пример создания собственных маршрутов.

Модуль 4

Контроллеры. Класс ActionResult

1. Контроллеры.
   - Создание нескольких методов действий в контроллере.
   - Обработка параметров методов действий в контроллере.
   - Добавление нескольких контроллеров.
   - Практические примеры использования.
2. ActionResult.
   - Что такое результат действия?
   - Цели и задачи класса ActionResult.
   - Потомки ActionResult:
     - ContentResult;
     - RedirectResult;
     - FileResult;
     - JavaScriptResult;

    - ViewResultBase;
    - другие потомки.
  - Практические примеры использования.
  - Селекторы:
    - что такое атрибут?
    - что такое селектор?
    - виды селекторов:
      - ActionName;
      - ActionVerbs;
      - NonAction.
    - практические примеры использования селекторов.
  - Фильтры:
    - что такое фильтр в ASP.NET MVC?
    - необходимость использования фильтров;
    - типы фильтров:
      - фильтры действий (IActionFilter);
      - фильтры авторизации (IAuthorizationFilter);
      - фильтры результата (IResultFilter);
      - фильтры исключений (IExceptionFilter).
    - фильтры действий:
      - OutputCache;
      - HandleError;
      - Authorize.
    - области накладывания фильтров;
    - создание собственного класса фильтра.

Модуль 5

Представления. Модели. Класс HtmlHelper. Движок Razor

1. Представление.
   - Класс представления.
   - Добавление представления.
   - Практические примеры использования.
2. Модель.
   - Добавление модели в проект.

- Настройка связи между моделью и контроллером.
- Настройка связи между моделью и видом.
- Практические примеры использования.

3. Класс HtmlHelper.
   - Цели и задачи класса HtmlHelper.
   - Методы класса HtmlHelper:
     - текстовые поля;
     - кнопки;
     - ссылки;
     - формы;
     - другие методы.
   - Практические примеры использования.
4. Razor view engine.
   - Что такое Razor view engine?
   - Базовые принципы работы Razor.
   - Понятие code expression.
   - Понятие блоков кода.
   - Использование видов.
   - Практические примеры.
   - Создание элементов управления:
     - создание элементов управления в формах;
     - принципы обработки элементов форм;
     - частичные виды (partial views).

Модуль 6

Использование источников данных в ASP.NET MVC приложениях

1. Связывание данных с моделью.
2. Использование стандартных механизмов ADO.NET в ASP.NET MVC.
3. Валидация данных модели.
4. Использование механизмов Scaffolding.
5. Использование Entity Framework.
6. Внедрение зависимостей в ASP.NET MVC приложения.

Модуль 7

Использование AJAX в ASP.NET MVC приложениях, безопасность ASP.NET MVC приложений

1. Использование AJAX в ASP.NET MVC приложениях.
2. Особенности настройки системы безопасности ASP.NET MVC приложений.
3. Настройка кэширования.
4. Практические примеры.

Модуль 8

Использование Bootstrap в ASP.NET MVC приложениях

1. Что такое Bootstrap?
2. Цели и задачи Bootstrap.
3. Подключение Bootstrap к проекту.
4. Анализ функциональности Bootstrap.
5. Позиционирование элементов.
6. Компоненты.
   - Что такое компонент?
   - Виды компонентов.
   - Подключение и настройка компонентов.
7. Использование таблиц и форм.
8. Практические примеры использования.

Модуль 9

Web API

1. Что такое Web API?
2. Цели и задачи Web API.
3. Использование Web API для создания сервиса.
4. Практические примеры создания веб-сервисов с помощью Web API.

Модуль 10

ASP.NET Core

1. Что такое ASP.NET Core?
2. Цели и задачи ASP.NET Core.
3. Преимущества ASP.NET Core.
4. Создание первого проекта на ASP.NET Core.
5. Анализ структуры ASP.NET Core проекта.
6. Конфигурирование ASP.NET Core проекта.
7. Middleware и ASP.NET Core.
8. MVC-архитектура и ASP.NET Core.
9. Использование движка Razor.
10. Использование ASP.NET Core Identity framework.

Модуль 11

Экзамен

ШАГ компьютерная АКАДЕМИЯ

Программа курса «Создание веб-приложений с использованием Angular и React»

Створення веб-додатків з використанням Angular і React
Development of Web Applications Using Angular and React

Цель курса

Обучить студента разработке клиентских сценариев с использованием фреймворков Angular и React. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

По окончании курса слушатель будет:

- владеть основами с взаимодействия с NodeJS;
- уметь производить сборку проекта с помощью различных инструментов;
- разбираться в тонкостях использования Webpack;
- понимать структуру Angular приложения;
- уметь применять правильные конструкции Angular в зависимости от поставленной задачи;
- уметь производить связывание данных в Angular приложении;
- применять Dependency Injection;
- разбираться в тонкостях реализации React приложений;
- уметь использовать Flux;
- уметь использовать Redux;
- взаимодействовать с формами.

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. NodeJS

Модуль 2. Сборка проектов с помощью Webpack и других инструментов

Модуль 3. Angular

Модуль 4. React

Модуль 5. Экзамен

Модуль 1

NodeJS

1. Что такое NodeJS?
2. Цели и задачи NodeJS.
3. Настройка окружения.
4. Инструменты для разработки и отладки.
5. Что такое модуль?
6. Структура приложения.
7. Менеджер пакетов npm.
8. Создание и регистрация собственных пакетов.
9. Работа с событиями.
10. Использование файловой системы.
11. Модули по взаимодействию с HTTP и URL.

Модуль 2

Сборка проектов с помощью Webpack и других инструментов

1. Что такое сборка проектов?
2. Проблемы, возникающие при сборке проекта.
3. Инструменты для сборки проектов.
   - Обзор инструментов.
   - Gulp.
   - Grunt.
   - Webpack.
   - Другие.
4. Webpack.
   - Что такое Webpack?
   - Цели и задачи Webpack.
   - Преимущества использования Webpack.
   - Актуальные версии Webpack.

- Основы использования.
- Простая сборка.
- Сборка нескольких скриптов.
- Подключение библиотек.
- Стили и файлы.
- Использование Webpack Dev Server.
- Практические примеры использования.

Модуль 3

Angular

1. Что такое Angular?
2. Цели и задачи Angular.
3. Понятие одностраничного приложения (SPA).
4. Анатомия Angular приложения.
5. Настройка окружения.
6. Понятие модуля.
7. Компонент.
   - Что такое компонет.
   - Создание класса компонента.
   - Практические примеры использования.
8. Шаблоны.
   - Что такое шаблоны?
   - Создание шаблонов.
   - Практические примеры использования.
9. Интерполяция.
10. Использование директив.
11. Data binding в Angular.
12. Pipes в Angular приложении.
13. Сервисы.
14. Dependency Injection.
    - Что такое Dependency Injection?
    - Цели и задачи Dependency Injection.
    - Примеры использования.

15. Использование HTTP запросов.
16. Навигация и пересылка данных.
17. Практические примеры использования.

Модуль 4

React

1. Что такое React?
2. Цели и задачи React.
3. Анатомия React приложения.
4. Настройка окружения.
5. Что такое Flux, Reflux?
6. Что такое Redux?
7. Что такое JSX?
8. Компоненты.
   - Что такое компонент?
   - Создание компонентов.
   - Практические примеры использования.
9. Жизненный цикл приложения на React.
10. Props и State.
11. Controller views.
12. Route.
13. Использование форм.
14. Использование Flux, Reflux.
15. Использование Redux.
16. Практические примеры использования.

Модуль 5

Экзамен

Программа курса
«Использование Microsoft Azure при разработке приложений»

Використання Microsoft Azure при розробці додатків
Microsoft Azure: a platform for application development

Цель курса:

Обучить слушателя основам использования и создания Microsoft Azure решений. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

По окончании курса слушатель будет:

- понимать архитектуру Microsoft Azure;
- уметь создавать, компилировать, и отлаживать проекты Microsoft Azure в Microsoft Visual Studio;
- разбираться в типах облачных решений;
- разбираться в особенностях настройки Azure-решений;
- понимать принципы взаимодействия с Azure Storage;
- уметь использовать SQL Azure;
- уметь мигрировать существующие базы данных в SQL Azure;
- использовать Service Fabric и Service Bus;
- применять когнитивные сервисы;
- знать основы использования машинного обучения.

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.

Перед началом данного предмета необходимо предоставить студентам доступ к следующему курсу Microsoft Imagine Academy:

- Data Science and Machine Learning Essentials.

Тематический план

Модуль 1. Облачные вычисления, облачные решения, облако. Введение в Microsoft Azure

Модуль 2. Создание программного проекта с использованием Microsoft Azure

Модуль 3. SQL Azure

Модуль 4. Service Fabric. Создание решений

Модуль 5. Когнитивные сервисы

Модуль 6. Использование машинного обучения

Модуль 7. Экзамен

Модуль 1

Облачные вычисления, облачные решения, облако. Введение в Microsoft Azure

1. Что такое облако?
2. Что такое облачные вычисления?
3. Что такое облачные решения.
4. Что такое:
   - Infrastructure as a Service;
   - Software as a Service;
   - Platform as a Service;
   - Desktop as a Service;
   - Backend as a Service.
5. Игроки на рынке облачных решений:
   - Amazon;
   - Microsoft;
   - Google;
   - Salesforce;
   - другие игроки.
6. Что такое Microsoft Azure:
   - история создания;
   - цели и задачи.
7. Обзор Microsoft Azure:
   - настройка Microsoft Azure;
   - Windows Azure Storage;
   - Worker Role;
   - Virtual Machine Role;
   - Windows Azure AppFabric;
   - SQL Azure.
8. Установка ПО для разработки Azure решений.
9. Эмуляция облака.
10. Создание первого Azure решения.

Модуль 2

Создание программного проекта с использованием Microsoft Azure

1. Анализ типов проектов для Microsoft Azure.
2. Создание и анализ проектов для Microsoft Azure.
3. Azure management portal.
4. Публикация проекта.
5. Конфигурирование и обновление проектов.
6. Azure Storage:
   - что такое Azure Storage?
   - API для работы со Storage;
   - использование таблиц;
   - отправка сообщений с помощью очередей;
   - использование blob-хранилища.

Модуль 3

SQL Azure

1. Что такое SQL Azure?
2. Свойства SQL Azure.
3. Доступ к SQL Azure.
4. Создание базы данных в облаке.
5. Использование облачной базы данных.
6. Миграция существующей базы данных в облако.

Модуль 4

Service Fabric. Создание решений

1. Что такое Service Fabric?
2. Компоненты Service Fabric.
3. Что такое Service Bus?
4. Endpoints.
5. Bindings:
   - что такое Bindings?

- цели и задачи Bindings;
- практические примеры использования.

Модуль 5
Когнитивные сервисы

1. Что такое когнитивные сервисы?
2. Виды сервисов:
   - голосовые сервисы;
   - сервисы по работе с изображениями;
   - языковые сервисы;
   - поисковые сервисы.
3. Практические примеры использования.

Модуль 6
Использование машинного обучения

1. Что такое машинное обучение?
2. Машинное обучение и Azure.
3. Azure Machine Learning.
4. Azure Machine Learning Studio.
5. Azure Machine Learning Services.
6. Практические примеры использования Machine Learning.

Модуль 7
Экзамен

ШАГ компьютерная АКАДЕМИЯ

Программа курса
«Создание web–приложений, исполняемых на стороне сервера при помощи языка программирования PHP, СУБД MySQL и технологии Ajax»

Створення web-додатків, виконанних з боку сервера
за допомогою мови програмування PHP і технології AJAX

Developing Server-Side Applications Using PHP and AJAX

Цель курса

Обучить слушателя разработке веб-приложений на платформе PHP с использованием СУБД MySQL.

По окончании курса слушатель будет:

- разбираться в тонкостях построения веб-приложений использованием PHP;
- понимать особенности реализации механизмов ООП в PHP;
- обрабатывать и анализировать данные форм;
- использовать стандартные функции PHP;
- применять регулярные выражения;
- сохранять данные пользователя в файлах cookies;
- работать с механизмом сессий;
- взаимодействовать с источниками данных;
- внедрять AJAX в веб-приложения.

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 web-программирование на PHP

Модуль 2. Работа с формами, функции

Модуль 3. ООП, регулярные выражения

Модуль 4. Работа с файлами, ошибки и исключения, cookies и сессии

Модуль 5. Взаимодействие с СУБД MySQL

Модуль 6. Ajax и PHP. Использование frameworks, CMS и PHP

Модуль 7. Экзамен

Модуль 1

Введение в web-программирование на PHP

1. Введение в web-программирование и принципы работы web-приложений:
   - обоснование и истоки возникновения. Отличия серверного web-программирования от клиентского. Цели, задачи, направление развития, краткая история;
   - архитектура «клиент-сервер». Выполнение серверных сценариев. CGI. Взаимодействие с СУБД;
   - принципы и этапы загрузки web-страницы.
2. Обзор и сравнительный анализ существующих серверных web-технологий:
   - ASP.NET (ASP);
   - JSP (sun);
   - Ruby on Rails (RoR);
   - ColdFusion (www.cffaq.com);
   - Python.
3. Введение в PHP:
   - что такое PHP и область применения;
   - история возникновения PHP. Обзор релизов PHP. Отличия PHP5 и PHP6;
   - информационные ресурсы, документация.
4. Описание и обзор инструментария для разработчика: web-серверы, СУБД.
5. Подготовка инструментария:
   - установка и настройка web-сервера Apache под Windows;
   - обзор директорий и файлов установленного сервера (модули и т. д.);
   - конфигурационный файл httpd.conf;
   - создание виртуальных хостов;
   - установка PHP для web-серверов Apache, IIS и первичная настройка;
   - обзор директорий и файлов. Конфигурационный файл php.ini;
   - краткий обзор отличий установки под Windows и под UNIX;
   - обзор и сравнение редакторов кода PHP: Eclipse PDT, Zend Studio, PHP Expert Editor, DreamWeaver, PHPStorm.
6. Формы включения PHP кода внутрь страницы:
   - полные теги;

- короткие теги;
- теги ASP;
- теги script.

7. Разделение выражений. Комментарии.
8. Правила именования переменных. Константы.
9. Типы данных:
   - целочисленный, вещественный, логический;
   - строковый, массивы, объектный, ресурсный;
   - преобразование типов данных;
   - семейство функций is..() (is_int, is_float, ...).
10. Операторы и операнды:
    - арифметические и логические операторы;
    - операторы присваивания.
11. Условные конструкции:
    - конструкция if;
    - конструкция switch.
12. Циклические конструкции:
    - цикл while;
    - цикл do…while;
    - цикл for;
    - досрочное прерывание итераций: break, continue.
13. Массивы:
    - массивы в PHP. Особенности строения массивов;
    - ассоциативные и индексные (списки) массивы. Способы инициализации. Конструкция array;
    - одномерные и многомерные массивы. Отличия многомерных массивов от одномерных. Способы инициализации;
    - способы перебора массивов. Конструкции list, each. Цикл foreach.

Модуль 2

Работа с формами, функции

1. Работа с формами. Способы связывания XHTML-формы и PHP-скрипта.
   - работа форм. Описание протокола http;
   - методы GET и POST. Структура и назначение. Совмещение методов;

- глобальные переменные. Настройка файла php.ini. Суперглобальные массивы $_POST, $_GET, $_REQUEST;
- передача различных элементов форм в скрипт и их обработка;
- форма и ее обработчик в одном сценарии.

2. Функции:
   - синтаксис функций и примеры объявления;
   - передача параметров по значению и по ссылке;
   - области видимости переменных. Локальные и глобальные переменные;
   - статические переменные;
   - условные функции;
   - объявление функции внутри функции;
   - функции с параметрами по умолчанию;
   - функции с бесконечным количеством параметров;
   - использование функций func_num_args(), func_get_arg(), func_get_args():
     - анонимные функции.
3. Создание библиотек:
   - инструкция require;
   - инструкция include;
   - инструкция require_once;
   - инструкция include_once.
4. Функции для работы с массивами:
   - функции сортировки массива (sort, asort, ksort, krsort, ...);
   - поиск в массиве;
   - слияние и разделение массивов;
   - другие полезные функции для массивов (array_sum, array_fill, array_walk, array_unique, …).
5. Математические функции:
   - округление;
   - вычисление чисел с произвольной точностью;
   - генерация случайных чисел.
6. Функции для работы со временем:
   - получение и конвертация дат;
   - сравнение дат.
7. Функции для работы со строками:
   - вывод строк в браузер. Сериализация;

- поиск, замена, урезание и разбор строк;
- функции для работы с XHTML. Кодировки;
- другие функции. Шифрование.

Модуль 3

ООП, регулярные выражения

1. Объектно-ориентированное программирование в PHP:
   - основные концепции ООП (инкапсуляция, полиморфизм, наследование) и их реализация в PHP. Анализ отличий аппарата ООП в PHP4 и PHP5;
   - синтаксис объявления классов;
   - поля и методы класса. Свойства класса. Спецификаторы доступа. Константы. Типы, передаваемых в метод параметров. Тип возвращаемого значения из метода;
   - конструкторы и деструкторы;
   - клонирование объектов;
   - статические свойства и методы класса;
   - метод __toString();
   - наследование и перегрузка методов;
   - «волшебные» (magic) методы __sleep(), __wakeup() и др.;
   - оператор instanceof;
   - финальные классы и методы. Анонимные классы;
   - абстрактные классы и методы;
   - интерфейсы;
   - функции для работы с классами: class_exists(), get_class_methods(), get_class_vars(), get_declared_classes(), get_declared_interfaces(), get_object_vars(), get_parent_class(), is_a(), is_subclass_of(), method_exists().
2. Поддержка регулярных выражений в PHP:
   - системы регулярных выражений PERL и POSIX;
   - регулярные выражения PERL. Синтаксис PERL совместимых выражений. Примеры регулярных выражений;
   - функции для использования PERL совместимых регулярных выражений.

Модуль 4

Работа с файлами, ошибки и исключения, cookies и сессии

1. Принципы взаимодействия с файлами и директориями средствами PHP:
   - понятие текстового и двоичного файла;
   - открытие и закрытие файла. Прямая работа с файлами;
   - чтение данных из файла. Запись данных в файл;
   - позиционирование по файлу;
   - дополнительные функции для работы с файлами: filemtime(), filesize(), filetype(), ftruncate(), is_file(), is_writable(), basename();
   - копирование, удаление и переименование файлов;
   - функции для работы с директориями;
   - взаимодействие с операционной системой и использование PHP в командной строке;
   - конфигурация PHP для работы с загрузкой файлов на сервер;
   - простая загрузка. Использование массива $_FILES;
   - множественная загрузка.
2. Ошибки и исключения:
   - регулирование вывода сообщений об ошибках и настройка конфигурационного файла;
   - изменение стандартного обработчика ошибок. Оператор отключения сообщений об ошибках. Отладка скриптов;
   - исключения:
     - что такое исключение или исключительная ситуация?
     - конструкция try;
     - конструкция catch;
     - конструкция throw;
     - конструкция finally;
     - класс Exception;
     - интерфейс Throwable;
     - отлов нескольких исключительных ситуаций в одном catch;
     - исключения и функции.
3. Cookies:
   - сравнение подходов к хранению пользовательской информации. Анализ клиентского подхода (cookies) и серверного подхода (сессии);

- использование cookies. Установка cookies. Функция setcookie. Использование массива $_COOKIE;
- удаление cookies. Проверка поддержки cookies.

4. Сессии:
   - сессии. Два подхода к использованию сессий. Использование cookies, URL для хранения id сессии;
   - настройка файла php.ini для сессий. Функции для использования сессий. Глобальный массив $_SESSION.

Модуль 5

Взаимодействие с СУБД MySQL

1. Обзор возможностей PHP по поддержке работы с разными СУБД.
2. Административные возможности СУБД MySQL:
   - история развития и инсталляция СУБД MySQL;
   - основы взаимодействия с СУБД MySQL. Отличия различных веток MySQL. Особенности диалекта SQL;
   - приложение phpMyAdmin;
   - поддержка расширенных конструкций SQL. Представления. Хранимые процедуры;
   - функции. Триггеры. Встроенные функции MySQL;
   - управление пользовательскими учетными записями в СУБД MySQL;
   - файлы журналов;
   - поддержка и восстановление баз данных. Проверка таблиц на наличие ошибок;
   - резервирование и копирование баз данных. Методы резервирования. Восстановление таблиц.
3. Взаимодействие PHP и MySQL:
   - библиотека mysql. Функции: mysql_connect(), mysql_pconnect(), mysql_select_db(), mysql_close(), mysql_query(), mysql_result(), mysql_db_name(), mysql_errno(), mysql_error(), mysql_fetch_array(), mysql_fetch_assoc(), mysql_fetch_row(), mysql_field_flags(), mysql_field_len(), mysql_field_type(), mysql_free_result(), mysql_list_dbs(), mysql_list_fields(), mysql_list_tables(), mysql_num_fields(), mysql_num_rows();
   - библиотека mysqli;

- процедурный стиль работы с mysqli;
- объектный стиль работы с mysqli.

4. Библиотека SQLite.
5. Расширение PHP Data Objects.
6. Работа с графикой:
   - графическая библиотека GD. Обзор ее возможностей и недостатков;
   - функции для создания, удаления и модификации изображений;
   - функции для получения информации о изображениях;
   - функции для рисования геометрических фигур;
   - функции для работы с текстом и шрифтами;
   - функции для работы с цветом;
   - функции для работы с пикселями.

Модуль 6

Ajax и PHP. Использование frameworks, CMS и PHP

1. Что такое Ajax?
2. Цели и задачи Ajax.
3. Объект XMLHttpRequest.
4. Пример использования XMLHttpRequest.
5. Библиотеки для работы с Ajax:
   - Sajax6;
   - AjaxAC;
   - Cajax;
   - другие библиотеки.
6. Практические примеры использования Ajax.
7. PHP frameworks. Обзор существующих решений:
   - Zend Framework;
   - Symfony4;
   - CakePHP;
   - CodeIgniter:
     - обзор Codeigniter;
     - взаимодействие контроллеров, моделей и представлений в Codeigniter;
     - принципы разработки сайта в CodeIgniter;
     - валидация данных форм в CodeIgniter;

- использование Query Builder;
- маршрутизация в CodeIgniter.

- Laravel:
  - обзор Composer;
  - обзор Artisan;
  - обзор структуры приложения Codeigniter;
  - создание контроллера;
  - создание представлений;
  - шаблонизатор Blade;
  - миграции;
  - работа с моделями ;
  - валидация данных форм в Laravel;
  - аутентификация в Laravel.

8. Web-мастеринг на основе CMS и обзор готовых проектов:
   - обзор рынка основных CMS, основанных на связке PHP и MySQL;
   - обзор устройства и принципов работы CMS WordPress;
   - процесс написания плагинов для WordPress;
   - обзор CMS Joomla!;
   - технология написания компонентов и плагинов для CMS Joomla!;
   - WYSIWYG редакторы для сайтов: FCKEditor, TinyMCE.

Модуль 7

Экзамен

Программа курса
«Программирование с использованием технологии Java и СУБД Oracle»

Програмування з використанням технології Java та СКБД Oracle
Programming using Java and Oracle DBMS

Цель курса

Обучить слушателя языку программирования Java и разработке серверных решений с использованием 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;
- 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;
- 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;
- использовать механизмы Spring и 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 язык программирования Java, переменные, типы данных, операторы, логические операторы, операторы ветвлений, побитовые операторы, циклы, строки, массивы, методы

Модуль 2. Объектно-ориентированное программирование, исключения

Модуль 3. JavaCollectionFramework

Модуль 4. Аннотации, Анонимные классы, Lambda-выражения

Модуль 5. Работа с файлами

Модуль 6. Многопоточность

Модуль 7. Сетевое взаимодействие

Модуль 8. Stream API

Модуль 9. Использование jUnit

Модуль 10. Использование СУБД Oracle

Модуль 11. Введение в разработку серверных решений с использованием Java

Модуль 12. Взаимодействие с источниками данных

Модуль 13. JavaServer Pages

Модуль 14. Tags в JSP

Модуль 15. Введение в Spring

Модуль 16. Введение в Hibernate, Spring Data

Модуль 17. Экзамен

Модуль 1

Введение в язык программирования Java, переменные, типы данных, операторы, логические операторы, операторы ветвлений, побитовые операторы, циклы, строки, массивы, методы

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. Программная среда Eclipse:
   - инсталляция;
   - основы работы с IDE Eclipse;
   - создание проекта;
   - добавление файла к проекту;
   - обзор альтернативных средств разработки;
   - запуск простейшего приложения.
3. 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.
4. Переменная:
   - необходимость использования переменных;
   - идентификаторы;
   - ключевые слова;
   - синтаксис объявления переменных;
   - константы и литералы;

- необходимость применения;
- синтаксис объявления.

5. 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.
6. Преобразование типов данных:
   - необходимость использования;
   - неявное преобразование типов;
   - явное преобразование типов.
7. Логические операторы:
   - знакомство с логическими операциями;
   - таблица результатов применения логических операций;
   - «логическое отрицание». Оператор!;
   - «логическое И». Оператор &&;
   - «логическое ИЛИ». Оператор ||.
8. Таблица приоритетов операторов.
9. Конструкции логического выбора. Операторы ветвлений:
   - оператор ветвления if;
   - оператор ветвления if-else;
   - лестница if-else if;

- понятие составного оператора;
- тернарный оператор;
- оператор множественного выбора – switch.

10. Побитовые операторы.
    - системы исчисления двоичная, восьмеричная, шестнадцатеричная;
    - цели и задачи битовых операций;
    - битовое «И»;
    - битовое «ИЛИ»;
    - битовое «ИСКЛЮЧАЮЩЕЕ ИЛИ»;
    - битовое отрицание;
    - битовые сдвиги.
11. Циклы:
    - необходимость использования циклов. Примеры использования;
    - цикл while;
    - цикл for;
    - цикл do-while;
    - операторы break и continue;
    - вложенные циклы. Примеры использования.
12. 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).
13. Работа со строками.
14. Массивы:
    - что такое массивы;
    - необходимость использования массивов;
    - синтаксис объявления одномерного массива;
    - схема размещения массивов в памяти;
    - индексация элементов массива.
15. Многомерные массивы:
    - многомерные массивы. Цели и задачи их использования;
    - двумерные массивы, как частный случай многомерных;

- синтаксис объявления многомерного массива;
- примеры использования многомерных массивов.

16. 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).
17. Область видимости:
    - понятие области видимости;
    - примеры использования областей видимости.
18. Рекурсия.
19. JShell:
    - что такое JShell;
    - цели и задачи JShell;
    - примеры использования JShell.

Модуль 2

Объектно-ориентированное программирование, исключения

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.

19. Что такое исключительная ситуация?
20. Принципы обработки исключительных ситуаций.
21. Понятие checked и unchecked исключений:
    - что такое checked и unchecked исключения;
    - отличия и принципы использования.
22. Ключевое слово try.
23. Ключевое слово catch.
24. Ключевое слово throw.
25. Ключевое слово finally.
26. Подробности использования исключительных ситуаций.
27. Раскрутка стека вызовов.

Модуль 3

JavaCollectionFramework

1. Классы-обертки.
2. Введение в JCF:
    - причины создания;
    - обзор.
3. Интерфейсы JCF:
    - Collection;
    - Comparator;
    - Enumeration;
    - EventListener;
    - Iterator;
    - List;
    - ListIterator;
    - Map;

- Map.Entry;
- Observer;
- RandomAccess;
- Set;
- SortedMap;
- SortedSet.

4. Классы JCF:
   - AbstractCollection;
   - AbstractList;
   - AbstractMap;
   - AbstractSequentialList;
   - AbstractSet;
   - ArrayList;
   - Arrays;
   - BitSet;
   - Collections;
   - Dictionary;
   - HashMap;
   - HashSet;
   - Hashtable;
   - IdentityHashMap;
   - LinkedHashMap;
   - LinkedHashSet;
   - LinkedList;
   - Stack;
   - TreeMap;
   - TreeSet;
   - Vector.

Модуль 4

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 5

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 6

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 7

Сетевое взаимодействие

1. Обзор пакета java.net.
2. Класс InetAddress.
3. Класс Socket.
4. Класс ServerSocket.
5. Класс DatagramSocket.
6. Класс DatagramPacket.
7. Практическая работа. Создание файлового сервера.

Модуль 8

Stream API

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

Модуль 9

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 10

Использование СУБД Oracle

1. История СУБД Oracle.
2. Архитектура СУБД Oracle.
3. Версии СУБД Oracle.
4. Утилиты:
   - SQL Plus;
   - Database Configuration Assistant;
   - Administration Assistant for Windows;
   - Net Configuration Assistant.
5. Демонстрация: Инсталляция СУБД Oracle.
6. Архитектура БД под управлением Oracle. Сравнение с базами данных других СУБД:
   - создание базы данных с помощью Database Configuration Assistant;
   - создание базы данных с помощью файла конфигурации.
7. Демонстрация:
   - создание базы данных и управления базами данных с помощью Database Configuration Assistant;
   - создание базы данных с помощью файла конфигурации.
8. Практическая работа: Создание базы данных с помощью файла конфигурации.
9. Обзор типов данных СУБД Oracle.
10. Создание структуры учебной базы данных с использование СУБД Oracle.
11. Написание запросов для выборки данных (Соединение таблиц, функции агрегирования, сортировка данных).
12. Представления:
    - синтаксис создания представлений;
    - примеры использования.
13. Триггеры:
    - синтаксис создания триггеров;
    - примеры использования.
14. Хранимые процедуры:
    - синтаксис создания хранимых процедур;
    - примеры использования.

Модуль 11

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;
   - какие 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 12

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 13

JavaServer Pages

1. Что такое JSP?
2. Цели и задачи JSP.
3. История возникновения JSP.
4. Понятие директивы.
5. Обработка ошибок в JSP.
6. Model View Controller:
   - что такое Model View Controller;
   - цели и задачи Model View Controller;
   - примеры создания серверных решений с помощью MVC.
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.

Модуль 14

Tags в JSP

1. Java Standard Tag Library:
   - что такое Java Standard Tag Library;
   - цели и задачи Java Standard Tag Library;
   - понятие Tag.
2. Различные виды Tags:
   - Core Tags;
   - Formatting Tags;
   - SQL Tags;
   - XML Tags;
   - JSTL functions.
3. Использование Conditional Tags.
4. Использование Iteration Tags.
5. Примеры использования других Tags.
6. Что такое Custom Tags?
7. Что такое Tag Files?
8. Что JSP Fragment?
9. Примеры использования.

Модуль 15

Введение в Spring

1. Что такое Spring?
2. Цели и задачи Spring.
3. История возникновения.
4. Архитектура Spring.
5. REST и SOAP:
   - что такое REST;
   - что такое SOAP;
   - практические примеры.
6. Spring MVC.
7. 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 16

Введение в Hibernate, Spring Data

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

Модуль 17

Экзамен

Программа курса
«Программирование мобильных приложений под платформу Android»

Програмування мобільних додатків під платформу Android
Android Application Development

Цель курса

Обучить слушателя основам разработки приложений под мобильные устройства на основе Android с использованием языка программирования Java. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

По окончании курса слушатель будет:

- понимать архитектуру Android;
- уметь создавать, компилировать, и отлаживать проекты в Android Studio;
- разбираться в принципах жизненного цикла приложений Android;
- использовать различные разновидности Activity;
- разрабатывать виджеты различной степени сложности;
- применять механизмы оповещений и диалоговые окна;
- использовать различные виджеты в зависимости от поставленной задачи;
- уметь создавать приложения, реагирующие на жесты;
- разбираться в механизмах графического вывода и работы с изображениями;
- взаимодействовать с источниками данных;
- владеть механизмами интеграции с таким веб-сервисами, как Facebook, Twitter;
- понимать принципы и требования к регистрации приложения в Google Play и других магазинах приложений.

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 Android. Основы Android. Установка необходимого ПО

Модуль 2. Структура android-проекта. Пользовательский интерфейс приложения

Модуль 3. Material Design

Модуль 4. Layout, Views

Модуль 5. Детальный обзор виджетов Android. Основные и полезные методы виджетов, важные аспекты их создания и работы

Модуль 6. Расширенные приемы работы с виджетами

Модуль 7. Меню, управляющая и оповещающая информация, диалоги

Модуль 8. Фрагменты

Модуль 9. Изображения, графика, анимация

Модуль 10. Сервисы

Модуль 11. Сохранение информации в телефоне

Модуль 12. Использование баз данных в Android-приложениях

Модуль 13. Адресная книга, календари, события, звонки, sms

Модуль 14. Асинхронность, AsyncTask

Модуль 15. Сетевые возможности, доступ к веб-сервисам

Модуль 16. Регистрация в Google Play

Модуль 1

Введение в Android. Основы Android. Установка необходимого ПО

1. Введение в Android:
   - что такое Android?
   - что такое Java под Android?
   - виртуальная машина;
   - кратко об архитектуре Android;
   - Android SDK.
2. Программная среда Android Studio, Android SDK:
   - инсталляция;
   - основы работы с Android Studio;
   - создание проекта;
   - добавление файла к проекту;
   - обзор альтернативных средств разработки;
   - что такое Android SDK?
   - особенности различных версий Android SDK;
   - запуск простейшего приложения:
     - запуск на устройстве;
     - запуск на эмуляторе.
3. Обзор общих сведений о платформе Android.
4. Уровни архитектуры Android.
5. Процесс выполнения кода. Виртуальная машина Dalvik. Виртуальная машина ART.
6. Типы Android приложений.
7. Установка необходимого ПО для разработки.
8. Настройка эмуляторов.
9. Детальный обзор для разработки Android приложений:
   - основные свойства;
   - популярные «горячие клавиши».

Модуль 2

Структура android-проекта. Пользовательский интерфейс приложения

1. Создание «Hello, World»-проекта.
2. Физическая структура проекта:
   - файл манифеста. Файл AndroidManifest.xml;
   - разметка. Файл разметки;
   - строки. Файл со строками;
   - файл R.java;
   - файл с исходным кодом.
3. Запуск проекта:
   - запуск проекта на эмуляторе;
   - запуск проекта на телефоне.
4. События:
   - что такое событие?
   - примеры событий;
   - создание простейшего обработчика события.
5. Жизненный цикл приложения Android.
6. Понятие Activity.
7. Жизненный цикл Activity.
8. Основные методы, реализующие цикл.
9. Намерения intent:
   - что такое намерение?
   - цели и задачи намерений;
   - понятие группы намерений;
   - создание intent;
   - запуск нового activity с передачей intent.

Модуль 3

Material Design

1. Что такое Material Design?
2. Цели и задачи Material Design.

3. Принципы Material Design.
4. Практические примеры приложений, использующих Material Design.

Модуль 4
Layout, Views

1. Цели и задачи разметки (layout).
2. View и ViewGroup:
   - что такое View?
   - что такое ViewGroup?
3. Виды разметки:
   - FrameLayout;
   - LinearLayout;
   - RelativeLayout;
   - GridLayout.
4. Примеры использования различных видов разметки.
5. Лучшие практики.
6. Виды и виджеты:
   - что такое вид?
   - что такое виджет?
   - обзор различных виджетов:
     - текстовые поля;
     - кнопки;
     - индикаторы;
     - дата и время;
     - другие виды.
7. Ресурсы:
   - что такое ресурсы?
   - классификация ресурсов:
     - строки;
     - цвет;
     - изображения;
     - другие ресурсы.
   - тема и стиль;
   - использование квалификаторов;
   - произвольные ресурсы.

Модуль 5

Детальный обзор виджетов Android. Основные и полезные методы виджетов, важные аспекты их создания и работы

1. Доступ к виджетам.
2. Создание виджетов.
3. Инициализация виджетов.
4. Базовые виджеты:
   - TextView;
   - EditText;
   - Button;
   - RadioButton;
   - CheckBox;
   - ToggleButton;
   - ImageButton;
   - ProgressBar;
   - SeekBar;
   - RatingBar;
   - ImageView;
   - AnalogClock;
   - DigitalClock;
   - Chronometer;
   - другие виджеты.
5. Обработка воздействий пользователя на виджеты.
6. Виджеты для скроллинга.
7. Практические примеры.

Модуль 6

Расширенные приемы работы с виджетами

1. Расширенные текстовые поля:
   - AutoCompleteTextView;
   - MultiAutoCompleteTextView.

2. Адаптеры:
   - что такое адаптер?
   - стандартные адаптеры:
     - класс ArrayAdapter;
     - класс BaseAdapter;
     - класс SimpleCursorAdapter.
3. Использование виджетов:
   - ListView;
   - Spinner;
   - Gallery;
   - GridView.
4. Практические примеры.

Модуль 7

Меню, управляющая и оповещающая информация, диалоги

1. Меню приложения. Инициализация и обработка информации.
2. Использование меню в приложениях.
3. Оповещение Notification. Создание, поддержка, обработка выбора, закрытие.
4. Диалоговые окна:
   - класс AlertDialog. Различные варианты использования AlertDialog;
   - стандартные диалоги:
     - DatePickerDialog;
     - TimePickerDialog;
     - ProgressDialog;
     - CharacterPickerDialog.
   - создание собственных диалогов:
     - различные способы создания собственных диалогов;
     - обмен данными.
5. Практические примеры.

Модуль 8

Фрагменты

1. Что такое фрагмент?
2. Цели и задачи фрагментов.
3. Создание фрагмента.
4. Жизненный цикл фрагмента.
5. Добавление фрагмента.
6. Удаление фрагмента.
7. Замена фрагмента.
8. Примеры создания и использования фрагментов.

Модуль 9

Изображения, графика, анимация

1. Работа с изображениями как с ресурсами.
2. Работа с изображениями как с внешними файлами.
3. Отрисовка графических примитивов.
4. Использование шрифтов.
5. Создание nine-patches.
6. Обзор asset studio.
7. Анимация:
   - виды анимации:
     - Tweened View Animations:
       - AlphaAnimation;
       - RotateAnimation;
       - ScaleAnimation;
       - TranslateAnimation.
     - Frame Animations;
     - Interpolated Property Animations.
   - примеры создания и использование анимации.
8. Практические примеры.

Модуль 10

Сервисы

1. Службы:
   - что такое служба?
   - цели и задачи службы;
   - жизненный цикл службы.
2. Слушатели BroadcastReceiver:
   - что такое широковещательная передача?
   - что такое BroadcastReceiver?
   - цели и задачи BroadcastReceiver;
   - жизненный цикл;
   - регистрация BroadcastReceiver;
   - широковещательная передача событий с использованием намерений;
   - использование LocalBroadcastManager.
3. Практические примеры.

Модуль 11

Сохранение информации в телефоне

1. Файлы.
   - что такое файл в Android?
   - чтение данных из файла;
   - запись данных в файл:
     - запись файла в папку приложения;
     - запись файла в публичную папку.
   - удаление файлов созданных приложением.
2. Использование SharedPreferenses для сохранения информации:
   - что такое SharedPreferences?
   - цели и задачи SharedPreferences;
   - метод getSharedPreferences;
   - изменение объекта предпочтений;
   - использование XML-библиотеки для работы с предпочтениями:
     - преимущества использования XML-библиотеки;
     - создание экрана предпочтений;

- использование специализированных элементов управления в экранах предпочтений:
  - CheckBoxPreference;
  - EditTextPreference;
  - ListPreference;
  - MultiSelectListPreference;
  - RingtonePreference;
  - PreferenceCategory.

Модуль 12

Использование баз данных в Android-приложениях

1. База данных SQLite:
   - что такое SQLLite?
   - особенности SQLLite;
   - особенности использования SQL в SQLLite;
   - класс SQLiteOpenHandler:
     - создание базы данных;
     - открытие базы данных;
2. Курсор:
   - что такое курсор?
   - цели и задачи курсоров.
3. Получение данных из базы данных.
4. Добавление данных в базу данных.
5. Обновление данных в базе данных.
6. Удаление данных в базе данных.
7. Контент-провайдер:
   - что такое контент-провайдер?
   - цели и задачи контент-провайдера;
   - создание контент-провайдера с использованием базы данных в качестве источника данных;
   - создание контент-провайдера с использованием отличного от базы данных источника данных;
   - URI-адрес контент-провайдера.
8. Практический пример приложения, использующего источник данных.

Модуль 13

Адресная книга, календари, события, звонки, sms

1. Что такое адресная книга?
2. Использование адресной книги:
   - запрос на доступ к адресной книги;
   - получение информации из адресной книги;
   - вставка и редактирование информации в адресной книге;
   - поиск по адресной книге;
   - другие операции с адресной книгой.
3. Работа со звонками и sms.
4. Что такое календарь и событие?
5. Использование календаря:
   - получение списка календарей;
   - добавление события;
   - удаление события;
   - добавление будильников;
   - другие операции с календарем.
6. Практические примеры:
   - приложение «Список контактов»;
   - приложение «Будильник».

Модуль 14

Асинхронность, AsyncTask

1. Асинхронность:
   - что такое асинхронность?
   - цели и задачи асинхронности;
   - класс AsyncTask;
   - создание новых асинхронных задач;
   - запуск асинхронных задач.
2. Использование ExecutorService.
3. Практические примеры.

Модуль 15

Сетевые возможности, доступ к веб-сервисам

1. Синхронные и асинхронные запросы.
2. Интеграция с существующими веб-сервисами:
   - интеграция Facebook в приложение;
   - интеграция Twitter в приложение;
   - интеграция с другими сервисами.
3. Практические примеры.
4. Использование сторонних библиотек:
   - Volley;
   - Retrofit.

Модуль 16

Регистрация в Google Play

1. Что такое Google Play?
2. Категории приложений в Google Play.
3. Регистрация приложения в Google Play.
4. Обновление существующего в Google Play приложения.
5. Подпись и распространение приложений.
6. Интеграция рекламных баннеров в приложение. Роль и назначение рекламных баннеров на разных стадиях раскрутки приложения.
7. Сравнительный анализ других магазинов приложений (Amazon и т. д.).

ШАГ компьютерная АКАДЕМИЯ

Программа курса «Разработка игровых приложений с использованием Unity»

Розробка ігрових додатків з використанням Unity
Development of Game Applications Using Unity

Цель курса

Обучить слушателя основам разработки игр с использованием платформы Unity и языка программирования C#. Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

По окончании курса слушатель будет:

- понимать причины возникновения платформы Unity;
- оперировать базовыми терминами платформы Unity;
- уметь создавать и настраивать сцены;
- разбираться в тонкостях работы с редактором сцен;
- уметь настраивать освещение;
- уметь работать с шейдерами;
- уметь создавать обработчики различных событий;
- настраивать физику для игровых объектов;
- использовать 2D и 3D объекты для разработки игр;
- уметь настраивать анимацию для игровых объектов;
- понимать особенности кроссплатформенной разработки.

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. В качестве экзаменационного задания необходимо создать игру. Желательно разработку игры начинать с первых модулей данного курса.

Тематический план

Модуль 1. Введение в Unity

Модуль 2. Основы создания игр, события ввода

Модуль 3. Физика и игровые объекты

Модуль 4. Шейдеры

Модуль 5. Освещение, камеры

Модуль 6. Взаимодействие с 2D-графикой

Модуль 7. Использование 3D-графики

Модуль 8. Анимация

Модуль 9. Работа со сценами

Модуль 10. Использование аудио

Модуль 11. Кроссплатформенная разработка игр

Модуль 12. Экзамен

Модуль 1
Введение в Unity

1. Что такое Unity?
2. История развития Unity
3. Почему стоит использовать Unity?
4. Что такое сцена.
5. Что такое игровые объекты?
6. Редактор сцен.
   - Что такое редактор сцен?
   - Цели и задачи редактора сцен.
   - Основы взаимодействия с редактором сцен.
   - Что такое инспектор?
7. Что такое координаты?
8. Что такое виды?
9. Что такое освещение?
10. Что такое запеченное освещение?
11. Что такое материалы?
12. Что такое шейдеры?
13. Скрипты.
    - Что такое скрипты?
    - Языки написания скриптов.
    - Использование редактора для скриптов.
    - Использование Visual Studio для создания скриптов.
14. Asset.
    - Что такое asset?
    - Какие asset'ы бывают?
    - Использование asset'ов при разработке игр.
15. Горячие клавиши Unity.

Модуль 2

Основы создания игр, события ввода

1. Создание каркаса первой игры.
2. Анализ каркаса первой игры.
3. Обработка событий.
   - Что такое события?
   - Что такое обработчик события?
   - Типы событий.
   - Примеры обработки событий.
4. Обработка событий ввода.
   - Что такое событие ввода.
   - Обработка событий ввода.
   - Примеры обработки событий ввода.
5. Управление объектами сцены.
6. Использование векторов.
7. Использование текстур и материалов.

Модуль 3

Физика и игровые объекты

1. Физика и игровые объекты.
   - Что такое игровой объект?
   - Нужно ли применять физическую модель для игрового объекта?
   - Позиция игрового объекта.
   - Объект Rigidbody:
     - что такое Rigidbody?
     - цели и задачи Rigidbody;
     - свойства Rigidbody;
     - методы Rigidbody;
     - примеры использования.
2. Использование математики.
   - Нужно ли применять математику в Unity?
   - Объект Math:
     - что такое объект Math?

    - цели и задачи Math;
    - свойства Math;
    - методы Math.
3. Задержки и интервалы.
    - Что такое задержка?
    - Что такое интервал?
    - Примеры использования задержек и интервалов.

Модуль 4
Шейдеры

1. Что такое шейдеры?
2. Цели и задачи шейдеров.
3. Принципы работы с шейдерами.
4. Разработка шейдеров.
5. Standard Shader.

Модуль 5
Освещение, камеры

1. Что такое освещение?
2. Цели и задачи освещения.
3. Основы работы с освещением.
4. Cookies.
5. Использование теней.
6. Карты освещения.
7. Запеченное освещение.
8. Камера.
    - Что такое камеры?
    - Цели и задачи камер.
    - Примеры использования камер.

Модуль 6
Взаимодействие с 2D-графикой

1. Что такое 2D-графика?
2. Что такое спрайт?
3. Принципы работы с 2D-графикой в Unity.
4. Инструменты для работы с 2D-графикой в Unity.
5. Примеры взаимодействия с 2D-графикой.

Модуль 7
Использование 3D-графики

1. Что такое 3D-графика?
2. Что такое 3D-объект?
3. Инструменты для создания 3D-объектов.
4. Импорт 3D-объектов в Unity.
5. Динамическое создание 3D-объекта.
6. Mesh.
   - Что такое Mesh?
   - Объект Mesh.
   - Взаимодействие с Mesh.
   - Использование SkinnedMesh.
7. Примеры взаимодействия с 3D-графикой.

Модуль 8
Анимация

1. Что такое анимация?
2. Цели и задачи анимации.
3. Общие принципы использования анимация.
4. Инструменты для анимации.
5. Отличия в анимации для 2D и 3D.
6. Mecanim.
   - Что такое Mecanim?

- Цели и задачи Mecanim.
- Примеры использования.

7. Практические примеры работы с анимацией.

Модуль 9
Работа со сценами

1. Что такое сцена?
2. Цели и задачи сцен.
3. Создание сцен.
4. Переход между сценами.
5. Различные способы использования сцен.
   - Сцены как элемент уровня.
   - Сцены как элемент меню.
6. Практические примеры по работе со сценами.

Модуль 10
Использование аудио

1. Зачем использовать аудио при разработке игр?
2. Источники звуков в игре.
3. Поддерживаемые аудиоформаты.
4. Механизмы и свойства для настройки аудио в игре.
5. Практические примеры использования звуков и мелодий.

Модуль 11
Кроссплатформенная разработка игр

1. Что такое кроссплатформенная разработка?
2. Особенности кроссплатформенной разработки.
3. Обзор платформ.
   - iOS.
   - Android.
   - Windows.

- Apple Mac.
- Web Player.
- WebGL.

4. Особенности iOS разработки.
   - Основы разработки Unity-приложения под iOS.
   - Создание аккаунта разработчика.
   - Структура Unity-проекта под iOS.
   - Настройки iOS player.
   - Настройка производительности под iOS.
   - Примеры настройки приложения.
5. Особенности Android разработки.
   - Основы разработки Unity-приложения под Android.
   - Создание аккаунта разработчика.
   - Структура Unity-проекта под Android.
   - Настройки Android player.
   - Настройка производительности под Android.
   - Примеры настройки приложения.

Модуль 12

Экзамен

www.itstep.org