**Введение в программирование**

*Продолжительность курса – 11 пар*

**Версия 1.0.1**

**Цель курса**

Продолжить обучение слушателя основам программирования (начало было положено в курсе «Введение в программирование»). Научить студента использовать циклы, массивы, методы и другие базовые конструкции языков программирования.

**По окончании курса слушатель будет:**

* Использовать циклы, методы и другие базовые конструкции
* Понимать и использовать отладчик
* Разбираться в основах использования массивов
* Понимать принципы ООП
* Понимать отличие абстрактного класса от интерфейса
* Понимать принципы обработки исключений

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

*Модуль 1 (2 пары)*

# Циклы, использование отладчика

1. Введение в программирование
   1. Структура программы: инструкции, блок кода
   2. Типы данных, переменные
2. Условные операторы
   1. Условие if … else …
   2. Переключатель switch
   3. Тернарный оператор
3. Циклы
   1. Необходимость использования циклов. Примеры использования.
   2. Цикл while.
   3. Цикл for.
   4. Цикл do-while.
   5. Обозначение циклов в блок-схемах. Блок цикла.
   6. Операторы break и continue.
   7. Примеры построения программ с использованием циклов на языке блок-схем.
   8. Вложенные циклы. Примеры использования.
4. Работа с интегрированным отладчиком.
   1. Что такое отладчик. Цели и задачи отладчика.
   2. Запуск программы по шагам.
   3. Окна для работы с отладчиком. Окна переменных, локальных переменных, памяти.
   4. Исполнение одного шага.
   5. Установка точки останова (breakpoint).
   6. Установка умной точки останова (smart breakpoint).

*Модуль 2 (2 пары)*

# Массивы одномерные и многомерные

1. Массивы
   1. Что такое массивы. Необходимость их использования
   2. Синтаксис объявления одномерного массива
   3. Способы инициализации массива
   4. Схема размещения массивов в памяти
   5. Индексация элементов массива
   6. Примеры использования массивов на языке блок-схем
2. Многомерные массивы
   1. Многомерные массивы. Цели и задачи их использования
   2. Двумерные массивы, как частный случай многомерных
   3. Синтаксис объявления многомерного массива
   4. Примеры использования многомерных массивов

*Модуль 3 (2 пары)* **Методы**

1. Методы

1. Необходимость использования методов
2. Синтаксис объявления методов
3. Использование ключевого слова void при работе с методами
4. Вызов метода
5. Аргументы метода
6. Возврат значения из метода (оператор return)
7. Понятие области видимости.
8. Передача массива в метод
9. Перегрузка методов
10. Рекурсия

*Модуль 4 (2 пары)*

# Объектно-ориентированное программирование (Обзор возможностей аппарата ООП в Java)

1. Введение в объектно-ориентированное программирование
   1. Инкапсуляция
   2. Полиморфизм
   3. Наследование
2. Понятие класса
3. Понятие объекта
4. Понятие члена класса, поля класса, метода класса
5. Спецификаторы доступа
6. Конструкторы объекта 1. Что такое конструктор?
   1. Цели и задачи конструктора
   2. Примеры создания конструкторов
7. Ключевое слово this
8. Перегрузка методов и конструкторов
9. Статические методы классов
   1. Что такое статический метод класса?
   2. Отличие статического и обычного метода класса
   3. Примеры использования статических методов
10. Передача объектов в метод
11. Область видимости в методах классов
12. Наследование
    1. Спецификаторы доступа при наследовании
    2. Ключевое слово super
    3. Порядок вызова конструкторов
    4. Переопределение методов
    5. Динамическая диспетчеризация методов
    6. Абстрактный класс
13. Понятие интерфейса
    1. Что такое интерфейс?
    2. Реализация интерфейса
    3. Использование реализации интерфейса через ссылки
    4. Вложенные интерфейсы
    5. Переменные и интерфейсы
14. Вложенные классы
15. Ключевое слово final
    1. Использование final для классов
    2. Использование final для методов
16. Сборка мусора
    1. Что такое сборка мусора?
    2. Принцип работы сборщика мусора
    3. Что такое финализатор?
    4. Метод finalize
    5. Принципы создания финализатора
17. Пакеты
18. Шаблоны (Generics) 1. Что такое шаблоны?
    1. Цели и задачи шаблонов
    2. Шаблонные классы
    3. Шаблонные методы
    4. Шаблонные конструкторы
    5. Шаблонные интерфейсы
    6. Шаблоны и наследование

*Модуль 5 (1 пара)*

# Исключения

1. Что такое исключительная ситуация?
2. Принципы обработки исключительных ситуаций
3. Понятие checked и unchecked исключений 1. Что такое checked и unchecked исключения?

2. Отличия и принципы использования

1. Ключевое слово try
2. Ключевое слово catch
3. Ключевое слово throw
4. Ключевое слово finally
5. Подробности использования исключительных ситуаций
6. Раскрутка стека вызовов

*Модуль 6 (1 пара)*

**Обзор Java Collection Framework** 1. Классы-обертки 2. Введение в JCF:

* 1. Причины создания
  2. Обзор
  3. Обзор интерфейсов JCF:
  4. Collection
  5. Comparator
  6. Enumeration
  7. Iterator
  8. List
  9. ListIterator
  10. Map
  11. Map.Entry
  12. Set
  13. SortedMap
  14. SortedSet

4. Обзор классов JCF:

* 1. ArrayList
  2. Arrays
  3. Collections
  4. HashMap
  5. HashSet
  6. LinkedHashMap
  7. LinkedHashSet
  8. LinkedList
  9. Stack
  10. TreeMap
  11. TreeSet

*Модуль 7 (1 пара)*

# Регулярные выражения

1. Что такое регулярное выражение?
2. Цели и задачи регулярного выражения
3. Структура регулярного выражения
4. Специальные обозначения в регулярных выражениях
   1. Метасимволы
   2. Символьные классы
   3. Квантификаторы
   4. Группы
5. Примеры использования регулярных выражений