Язык программирования Java

Версия 2.0.0

Продолжительность курса – 80 пар

Цель курса

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

По окончании курса слушатель будет:

- Понимать фундаментальные принципы создания программ с использованием Java
- Уметь создавать, компилировать, и отлаживать проекты в IDE Eclipse
- Уметь проектировать и реализовывать различные алгоритмы
- Использовать механизмы условий и циклов
- Применять массивы для хранения данных
- Уметь использовать алгоритмы сортировки и поиска данных
- Разбираться в принципах ООП
- Уметь проектировать классы различной степени сложности
- Создавать иерархии классов для решения практических задач
- Использовать механизмы generics для построения шаблонных классов
- Уметь порождать и обрабатывать исключительные ситуации
- Выбирать и использовать классы JCF
- Сохранять и читать информацию из файлов
- Понимать механизмы многопоточности Java
- Использовать лямбды
- Уметь пользоваться системой контроля версий
- Понимать основы командного взаимодействия
- Применять паттерны проектирования
- Использовать юнит-тестирование

По окончании данного курса студент сдаёт 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.

Модуль 1 (4 пары)

Введение в язык программирования “Java”

1. Вступление
    1. История и этапы развития языка “Java”
    2. Сравнительный анализ языка “Java” с другими языками программирования
    3. Что такое виртуальная машина?
    4. Что такое байт-код?

2. Алгоритм
    1. Понятие алгоритма
    2. Примеры использования алгоритмов в реальной жизни
    3. Типы алгоритмов. Линейный, разветвлённый, циклический
3. Понятие блок-схемы
    1. Базовые обозначения в блок-схемах
    2. Блок начала алгоритма
    3. Блок завершения алгоритма
    4. Блок ввода данных
    5. Блок вывода данных
    6. Блок вычислений
    7. Простейшие примеры использования блок-схем
4. Программная среда “Eclipse”
    1. Инсталляция
    2. Основы работы с IDE Eclipse
    3. Создание проекта
    4. Добавление файла к проекту
    5. Обзор альтернативных средств разработки
    6. Запуск простейшего приложения
5. JShell
    1. Что такое JShell?
    2. Цели и задачи JShell
    3. Примеры использования JShell

Модуль 2 (4 пары)

Переменные, типы данных, операторы

1. Типы данных
    1. Понятие типа данных. Размер, диапазон значений
    2. Целые типы данных
    3. Типы данных для работы с дробными числами
    4. Символьный тип данных
    5. Логический тип данных
    6. Перечислимый тип данных
2. Переменная
    1. Необходимость использования переменных
    2. Идентификаторы
    3. Ключевые слова
    4. Синтаксис объявления переменных
3. Константы и литералы
    1. Необходимость применения
    2. Синтаксис объявления
4. Операторы
    1. Понятие оператор
    2. Типы операторов
        1. Арифметические операторы
        2. Логические операторы
        3. Операторы ветвлений
        4. Унарные операторы
        5. Бинарные операторы

6. Тернарный оператор
3. Оператор присваивания
4. Арифметические операторы
    1. Оператор сложения
    2. Оператор вычитания
    3. Оператор умножения
    4. Оператор деления
    5. Оператор деления по модулю
    6. Инкремент. Постфиксная и префиксная форма
    7. Декремент. Постфиксная и префиксная форма
    8. Сокращенные формы
5. Примеры построения программ с использованием блок-схем

Модуль 3 (5 пар)

Логические операторы, операторы ветвлений, побитовые операторы

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

Модуль 4 (5 пар)

Циклы

1. 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

2. Работа с интегрированным отладчиком в Eclipse
    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)

Модуль 5 (5 пар)

Строки, массивы одномерные, многомерные

1. Работа со строками
2. Массивы
    1. Что такое массивы?
    2. Необходимость использования массивов
    3. Синтаксис объявления одномерного массива
    4. Схема размещения массивов в памяти.
    5. Индексация элементов массива
    6. Примеры использования массивов на языке блок-схем
3. Алгоритмы суммирования
4. Алгоритмы поиска
    1. Линейный
    2. Бинарный
5. Алгоритмы сортировки
    1. Пузырьковая сортировка
    2. Сортировка выбором
    3. Сортировка вставками
    4. Другие алгоритмы сортировки (быстрая сортировка и т.д.)
6. Понятие сложности алгоритма
7. 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 6 (4 пары)

Методы (на примере статических методов)

1. Методы
    1. Что такое метод?
    2. Необходимость использования методов
    3. Синтаксис объявления методов
    4. Использование ключевого слова void при работе с методами

5. Вызов метода
6. Аргументы
7. Возврат значения из метода (return)

2. Область видимости
    1. Понятие области видимости
    2. Примеры использования областей видимости
3. Рекурсия

Модуль 7 (16 пар)

Объектно-ориентированное программирование

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?
    2. Цели и задачи конструктора
    3. 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
    6. Статические методы интерфейсов
    7. Приватные методы интерфейсов
    8. Методы по умолчанию
    9. Функциональные интерфейсы
        1. Что такое функциональные интерфейсы?
        2. Цели и задачи функциональных интерфейсов

3. Стандартные функциональные интерфейсы
    1. Предикаты
    2. Функции
    3. Поставщики
    4. Компараторы
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?
    2. Цели и задачи шаблонов
    3. Шаблонные классы
    4. Шаблонные методы
    5. Шаблонные конструкторы
    6. Шаблонные интерфейсы
    7. Шаблоны и наследование

Модуль 8 (4 пары)

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
4. Ключевое слово try
5. Ключевое слово catch
6. Ключевое слово throw
7. Ключевое слово finally
8. Подробности использования исключительных ситуаций
9. Раскрутка стека вызовов

Модуль 9 (4 пары)

JavaCollectionFramework

1. Классы-обертки
2. Введение в JCF:
    1. Причины создания
    2. Обзор
3. Интерфейсы JCF:
    1. Collection
    2. Comparator
    3. Enumeration

4. EventListener
5. Iterator
6. List
7. ListIterator
8. Map
9. Map.Entry
10. Observer
11. RandomAccess
12. Set
13. SortedMap
14. SortedSet

4. Создание коллекций с помощью фабричных методов
5. Классы JCF:
    1. AbstractCollection
    2. AbstractList
    3. AbstractMap
    4. AbstractSequentialList
    5. AbstractSet
    6. ArrayList
    7. Arrays
    8. BitSet
    9. Collections
    10. Dictionary
    11. HashMap
    12. HashSet
    13. Hashtable
    14. IdentityHashMap
    15. LinkedHashMap
    16. LinkedHashSet
    17. LinkedList
    18. Stack
    19. TreeMap
    20. TreeSet
    21. Vector

Модуль 10 (2 пары)

Аннотации, Анонимные классы, Lambda выражения

1. Аннотации
2. Анонимные классы
3. Lambda выражения
    1. Что такое лямбда-выражения?
    2. Цели и задачи лямбда-выражений
    3. Синтаксис лямбда-выражений
    4. Примеры создания лямбда-выражений

Модуль 11 (3 пары)

Работа с файлами

1. Знакомство с пакетом java.io

2. Потоки ввода/вывода
    1. Потоки ввода/вывода
    2. Фильтрованные потоки
    3. Канальные потоки
    4. Буферизированные потоки
    5. Файловые потоки
    6. Потоки для работы с файлами
    7. Потоки, размещаемые в оперативной памяти
3. Сериализация объектов
    1. Понятие сериализации
    2. Граф сериализации
    3. Использование сериализации

Модуль 12 (2 пары)
Stream API

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

Модуль 13 (3 пары)
Многопоточность
1. Многопоточность в Java
    1. Что такое многопоточность?
    2. Класс Thread
    3. Интерфейс Runnable
    4. Приоритеты потоков
    5. Синхронизация потоков
        1. Проблемы, возникающие при синхронизации потоков
        2. Метод wait
        3. Метод notify
        4. Метод notifyall
2. Использование ExecutorService
3. Практические примеры

Модуль 14 (3 пары)
Системы контроля версий
1. Что такое контроль версий?
2. Зачем нужен контроль версий
3. Обзор систем контроля версий
    1. CVS
    2. SVN
    3. Git
    4. Другие системы контроля версий
4. Git
    1. Что такое Git?
    2. Цели и задачи Git?
    3. Основные термины

1. Репозиторий
2. Коммит
3. Ветка
4. Рабочий каталог

4. Операции с Git
    1. Установка
    2. Создание репозитория
    3. Добавление файла в репозиторий
    4. Запись коммита в репозиторий
    5. Получение текущего состояния рабочего каталога
    6. Отображение веток
    7. Операции с накопительным буфером
    8. git remote
    9. git push
    10. git pull
    11. Другие операции
5. Использование внешних сервисов (github)

Модуль 15 (4 пары)

Работа в команде, управление программными проектами

1. Что такое управление программными проектами?
2. Причины возникновения дисциплины управление программными проектами.
3. Диаграммы Ганта.
4. Важные вопросы по управлению программными проектами
    1. Что такое проект и программный проект?
    2. Что такое жизненный цикл процесса разработки программного обеспечения?
    3. Что такое управление проектами?
    4. Что такое одиночная разработка?
    5. Что такое командная разработка?
    6. Анализ проблем одиночной и командной разработки программного обеспечения
5. Анализ терминов предметной области
    1. Процесс
    2. Проект
    3. Персонал
    4. Продукт
    5. Качество
6. Характеристики проекта
    1. Тип проекта
    2. Цель проекта
    3. Требования к качеству
    4. Требования к бюджету
    5. Требования по срокам завершения
7. Расходы, связанные с проектом
    1. Прямые
    2. Непрямые
8. Общий обзор моделей и методологий процесса разработки
    1. Фазы процесса

- Определение требований
- Проектирование
- Конструирование (“реализация”, “кодирование”)
- Интеграция
- Тестирование и отладка (“верификация”)
- Инсталляция
- Поддержка

2. Водопадная модель
3. Спиральная модель
4. Итеративная модель
   - Agile
   - Scrum
   - XP
5. RUP
6. MSF
7. Анализ существующих моделей и методов

9. 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
      - Практическое задание: Необходимо провести симуляцию работы команды по методологии Scrum. Например, это может быть так называемое скрам-лего. Подробно тут: https://www.scrumalliance.org/system/resource_files/0000/3689/Scrum-Simulation-with-LEGO-Bricks-v2.0.pdf

Модуль 16 (2 пары)

Использование jUnit

5. Что такое модульное тестирование?
6. Цели и задачи модульного тестирования
7. Необходимость модульного тестирования
8. Обзор инструментов для модульного тестирования
9. Инструмент jUnit
   1. Что такое jUnit?

2. История создания jUnit
3. Практические примеры использования jUnit

Модуль 17 (4 пары)

Паттерны проектирования

1. Что такое паттерны проектирования
2. Причины возникновения паттернов проектирования
3. Понятие паттерна проектирования
4. Принципы применения паттернов проектирования
5. Принципы выбора паттернов проектирования
6. Принципы разделения паттернов на категории
7. Введение в UML
    a. Диаграмма классов
    b. Диаграмма объектов
    c. Диаграмма взаимодействия
8. Использование UML при анализе паттернов проектирования
    a. Диаграмма классов
    b. Диаграмма объектов
    c. Диаграмма взаимодействия
9. Порождающие паттерны
    a. Что такое порождающий паттерн?
    b. Цели и задачи порождающих паттернов
    c. Обзор порождающих паттернов
    d. Разбор порождающих паттернов
        i. Abstract Factory
            1. Цель паттерна
            2. Причины возникновения паттерна
            3. Структура паттерна
            4. Результаты использования паттерна
            5. Практический пример использования паттерна
        ii. Builder
            1. Цель паттерна
            2. Причины возникновения паттерна
            3. Структура паттерна
            4. Результаты использования паттерна
            5. Практический пример использования паттерна
        iii. Factory Method
            1. Цель паттерна
            2. Причины возникновения паттерна
            3. Структура паттерна
            4. Результаты использования паттерна
            5. Практический пример использования паттерна
        iv. Prototype
            1. Цель паттерна
            2. Причины возникновения паттерна
            3. Структура паттерна
            4. Результаты использования паттерна
            5. Практический пример использования паттерна
        v. Singleton

1. Цель паттерна
2. Причины возникновения паттерна
3. Структура паттерна
4. Результаты использования паттерна
5. Практический пример использования паттерна

10. Структурные паттерны
    a. Что такое структурный паттерн?
    b. Цели и задачи структурных паттернов
    c. Обзор структурных паттернов
    d. Разбор структурных паттернов
        i. Adapter
        ii. Composite
        iii. Facade
        iv. Proxy
        v. Другие структурные паттерны
11. Паттерны поведения
    a. Что такое паттерны поведения?
    b. Цели и задачи паттернов поведения
    c. Обзор паттернов поведения
    d. Разбор паттернов поведения
        i. Command
        ii. Iterator
        iii. Observer
        iv. Strategy
        v. Другие структурные паттерны

Модуль 18 (2 пары)

Паттерн MVC

1. Что такое паттерн MVC?
2. Цели и задачи паттерна Model-View-Controller
3. Model
    a. Что такое Model?
    b. Цели и задачи Model
4. View
    a. Что такое View?
    b. Цели и задачи View
5. Controller
    a. Что такое Controller?
    b. Цели и задачи Controller
6. Примеры использования паттерна MVC

Модуль 19 (2 пары)

Принципы проектирования классов SOLID

1. Обзор проблем, встречающихся при проектировании и разработке классов
2. Принципы проектирования классов SOLID
    a. Принцип единственности ответственности (The Single Responsibility Principle)
    b. Принцип открытости/закрытости (The Open Closed Principle)

c. Принцип подстановки Барбары Лисков (The Liskov Substitution Principle)
d. Принцип разделения интерфейса (The Interface Segregation Principle)
e. Принцип инверсии зависимостей (The Dependency Inversion Principle)

3. Примеры использования принципов SOLID

Модуль 20 (2 пары)
Экзамен