1. ВВ
2. Core

Программа курса

ВВ :Общая характеристика языка Java

* + История языка Java. Байт-код. Виртуальная машина Java (JVM). Трансляция, интерпретация и компиляция. JIT-компиляция. JDK и его структура. Команды JDK: javac, java, appletviewer, jar (en10). Их использование и параметры. Поиск и загрузка классов. Создание и запуск первой приложения на Java. Выход из программы и завершения ее работы. Использование Eclipse и NetBeans IDE: создание и выполнение прикладных программ.

ВВ:Типы данных и основные операции с ними

* + Алфавит языка. Комментарии. Ключевые слова. Простые (примитивные) типы данных. Идентификаторы, выбор имен идентификаторов. Переменные и константы. Объявления и инициализация переменных. Типы данных: простые и по ссылке. Область видимости и время жизни переменных. Операторы. Операции: арифметические, присваивания, отношения и логические. Преобразование и приведение числовых типов. Приоритет операций. Использование круглых скобок.

ВВ: Поток управления

* + Составной оператор (блок). Условный оператор и его виды. Циклы: for, while, do while. Бесконечные циклы. Оператор switch. Прерывание потока управления, метки. Оператор goto.

ВВ: Массивы

* + Объявления и инициализация массивов, в т.ч. анонимных. Линейные и многомерные массивы, «неравные» массивы. Итерация по элементам массива. Массивы и присвоения. Передача и возврат массивов в / из методов. Копирование массивов. Класс Arrays: заполнение, сравнение элементов и их сортировка. Поиск в массивах. Параметры командной строки. Методы: аргументы переменной длины.

ВВ-Core Введение в ООП. Введение в UML.

* + Объектная модель:
    - Основные положения: абстрагирование, инкапсуляция, модульность, иерархия
  + Объекты и классы:
    - Объекты
    - Определение (с примерами)
    - Состояние
    - Поведение
    - Операции
    - Идентичность
    - Инкапсуляция

Core: Классы и интерфейсы:

* + - UML. НотацияUML. Диаграммы классов.
    - Отношение между объектами
    - Зависимости, их разновидности
    - Обобщение
    - Ассоциации
    - Реализации
    - Связи, их множественность

Core: Объекты и классы

* + Классы. Поля и методы (определение, синтаксис объявления). Спецификаторы доступа public и private
  + Поля классов: инициализация явная и по умолчанию, спецификатор final и константы. Доступ к полям и методам
  + Методы: параметры (ссылки и простые типы) и возвращаемые, использование setта get- методов, перезагрузки методов, правила перезагрузки, ключевое слово strictfp
  + Конструкторы: конструкторы по умолчанию, вызов одного конструктора из другого
  + Блоки инициализации
  + Статические поля и методы: инициализация константы, статические блоки инициализации. Использование this для доступа к членам класса
  + Сбор мусора и удаления объектов

Core: Наследование и полиморфизм. Пакеты.

* + Наследование и полиморфизм как инструмент создания ПО. Суперкласс и подклассы, отношение обобщения, наследование в Java. Вызов конструкторов суперклассов, порядок вызовов. Ключевое слово super ()
  + Переопределение методов, правила переопределения. Наследование и статические поля и методы классов. Абстрактные классы и методы, – понятие, вызов конструкторов, реализация методов в субклассы
  + Интерфейсы. «Множественное» наследование. Определение и реализация интерфейсов. Расширение интерфейсов. Композиция как альтернатива множественному наследованию
  + Полиморфизм. Иерархия наследования
  + Преобразование и приведение типов. Использование instanceof. Отказ от наследования: терминальные (final) методы и классы. Включение как альтернатива множественному наследованию
  + Пакеты. Вложенность пакетов. Повторное использование кода и пакеты. Защита доступа. Импорт: классов, статических методов и полей

Core: Класс Object

* + Глобальный суперкласс Object
  + Методы equals (), hashCode (), toString (), finalize (), clone () и др
  + Правила реализации equals () и hashCode ()
  + Интерфейс Cloneableта и его реализация
  + Клонирование: поверхностное и глубокое

Core: Внутренние классы. Перечисление

* + Синтаксис объявления
  + Создание объектов внутренних классов
  + Поля и методы внутренних классов
  + Вложенность внутренних классов и доступ к членам внешних классов
  + Ссылка this во внутреннем классе
  + Доступ к полям и методам внешнего класса с внутреннего
  + Внутренние классы и спецификаторы доступа
  + Вложенные интерфейсы
  + Применение внутренних классов
  + Внутренние локальные классы (Method-LocalInnerClasses)
  + Анонимные внутренние классы
  + Вложенные (внутренние статические) классы (staticnestedclasses)
  + Создание экземпляров
  + Доступ к экземпляру внешнего класса
  + Компиляция программ с внутренними классами
  + Перечисление. Перечисление как классы. Методы values (), valueof (), name (), ordinal (). Константы, конструкторы, методы и поля
  + Перечисление и конструкция switch. Импорт членов перечислений

ВВ:Презентация программного проекта.Строки

* + Класс String. Неизменные объекты
  + Хранение строк в памяти
  + Основные операции класса String (создание, конкатенация, сравнение, поиск символов и подстрок в строках и др.). Преобразование простых типов данных. Разбивка строк: класс StringTokenizer. Переменные объекты: классы StringBuffer и StringBuilder
  + Основные методы. StringBuffer vs StringBuilder
  + Классы BigInteger и BigDecimal. Основные операции
  + Документирование кода
  + Включение комментариев
  + Документирование пакетов
  + Дескрипторы общего назначения
  + Комментарии к классам, методам и полей
  + Использование javadoc

Core: Обработка ошибок и исключения

* + Подходы к обработке ошибок времени выполнения
  + Понятие исключения и исключительной ситуации
  + Иерархия классов исключений в Java
  + Контролируемые (что проверяются) и неконтролируемые (не проверяются) исключения
  + Генерация исключительной ситуации и оператор throw
  + Объявления контролируемых исключений, ключевое слово throws
  + throws и перезагрузки методов
  + Перехват исключений
  + Перехват «всех» исключений
  + Обработка исключений
  + Повторная генерация исключений
  + Блок finally: освобождение ресурсов
  + Создание собственной иерархии исключений
  + Регистрация ошибок, логирования: логеры, уровень логгер, уровень сообщений, обработчики (объекты подкласса java.util.logging.Handler)
  + Пример

Core: Reflection

* + Основные возможности Reflection API
  + Класс Class
  + Использование Reflection API для получения указателей на функции
  + Использование Reflection API для доступа к частным членам

Core: Система ввода-вывода

* + Потоки, иерархия классов потокового ввода-вывода: понятие потока, потоки ввода / вывода, механизм работы с потоками, иерархия потокового ввода-вывода
  + Субклассы FilterOutputStream и FilterInputStream
  + Байтовые потоки
  + Символьные потоки
  + Класс File. Класс RandomAccessFile
  + Сериализация и десериализация объектов: понятие, назначение, реализация
  + Пример
  + Интерфейс Externalizable

Core: Универсальные типы. Классы коллекций

* + Параметризованные типы: классы и интерфейсы. Параметры типа
  + Ограничения, наложенные на параметризованные типы: простые типы, инстанциирование, суперкласс, исключение. Параметризованные типы и JVM
  + Понятие абстрактного типа данных (АТД)
  + Структуры данных: списки, стеки, множества, дерева, карты
  + Обход коллекций: интерфейсы Iteratorта и ListIterator, цикл «foreach»
  + Компараторы
  + Класс Collections: поиск и сортировка элементов, заполнения, синхронизован представления, неизменные представления и др

Core: Многопоточность

* + Понятие потока. Состояние потока: создание потоков, -runnable и -runninig – потоки, waiting / blocked / sleeping-потоки, состояние потока dead
  + Приоритеты потоков. Метод yield ()
  + Метод join (). Потоки-демоны
  + Синхронизация: синхронизация методов, Синхронизационные блоки. Получение lock объекта и блокировки. Взаимодействие потоков (ThreadInteraction)

Core: Работа с базами данных

* + Основы построения баз данных. Нормальные формы. Запросы SELECT, INSERT, UPDATE, DELETE

Core: JDBC

* + Driver, Connection, Statement, ResultSet. Пул соединений. Data Access Object

Core: Работа с XML

* + Знакомство с XML. Понятие формальной корректности и допустимости. Потоковый анализ на основе SAX. Анализ на основе DOM. Генерация Java-классов по схеме

Core: Новое в Java8

Core: Сетевые возможности Java

* + Класс URL. Класс URLConnection. Сокеты

Core: Технология RMI

* + Общие понятия о технологиях распределенного программирования. Принципы организации RMI

Core: Java Beans

* + Понятие о Java Beans
  + Проблема persistence. Сохранение бина в XML-файле

Core: Основы веб-программирования

* + Принципы клиент-серверного взаимодействия
  + Понятие и жизненный цикл сервлета
  + Организация JSP-страницы
  + Предопределено переменные
  + Взаимодействие JSP та JavaBeans
  + Передача данных между страницами
  + Область видимости бина

Core: Inversion of Control в Java.

* + Понятие Inversion of Control
  + Характеристика SpringIOC
* Контрольный тест
* Защита программного проекта

После изучения курса вы сможете:

* Разрабатывать, проводить отладку приложений и внедрять аппликации в рамках платформы JavaSE
* Быть знакомым с основными элементами технологии и конструкциями языка JavaSE
* Знать основные архитектурные и проектировочные приемы в рамках платформы
* Также курс предусматривает самостоятельное написание практического программного проекта под руководством преподавателя с его последующей презентацией

\*Объявленный формат чтения занятий предварительный. За уточнениями обращайтесь, пожалуйста, к менеджеру.