# Тема

«Введное занятие»

# План занятия

## История Java

**Основные теги**: проект Green, язык OaK, помощник Duke, 1st Person, неудача идеи интерактивного телевидения, www, Mosaic 1.0, WebRunner, 95 год и переименование, HotJava, Netscape Navigator 2.0, JavaSoft

## Платформа Java

**Основные теги**: JVM , кроссплатформенность, компилируемые и интерпретируемые языки, объектная ориентированность, строгая типизация, GC, безопасность, основные версии, JIT, JRE, три направления, Java HotSpot

## Методология процедурно-ориентированного программирования

**Основные теги**: Алгоритм, блок-схема, *нисходящее проектирование*, измерение сложности программ и продуктивности программистов

## Методология объектно-ориентированного программирования

**Основные теги**: Развитие PC, моделирование предметной области, формализация моделей

### Объекты

*Объект* - это мыслимая или реальная сущность, обладающая характерным поведением и отличительными характеристиками и являющаяся важной в предметной области

#### Состояние

*Состояние* - совокупный результат поведения объекта: одно из стабильных условий, в которых объект может существовать, охарактеризованных количественно; в любой момент времени состояние объекта включает в себя перечень (обычно статический) свойств объекта и текущие значения (обычно динамические) этих свойств

#### Поведение

*Поведение* - действия и реакции объекта, выраженные в терминах передачи сообщений и изменения состояния, видимая извне и воспроизводимая активность объекта

#### Уникальность

*Уникальность* - свойство объекта; то, что отличает его от других объектов

### Классы

*Класс* - это шаблон поведения объектов определенного типа с заданными параметрами, определяющими состояние.

UML представление

#### Инкапсуляция

*Инкапсуляция* (encapsulation) - это сокрытие реализации класса и отделение его внутреннего представления от внешнего (интерфейса).

*Модульность –* возможностьменять внутреннюю логику отдельного *класса*, не меняя код остальных компонентов системы.

#### Наследование

*Наследование* (inheritance) - это отношение между классами, при котором класс использует структуру или поведение другого класса (одиночное наследование ), или других (множественное наследование ) классов.

#### Полиморфизм

*Полиморфизм* (polymorphism) - положение теории типов, согласно которому имена (например, переменных) могут обозначать объекты разных (но имеющих общего родителя) классов. Следовательно, любой объект, обозначаемый полиморфным именем, может по-своему реагировать на некий общий набор операций

### Типы отношений между классами

#### Агрегация

#### Ассоциация

#### Наследование

#### Метаклассы

### Достоинства ООП

Высокая масштабируемость кода

Сокращение времени на разработку за счет использования компонент

Повышение качества за счет применения стабильных компонент

### Недостатки ООП

Требуется более серьезное документирование

Разделение кода по иерархии наследования усложняет понимание

Работает медленнее, чем процедурное

## Лексика языка

### Кодировка

**Основные теги**: UTF 16, native2ascii

### Анализ программы

#### Пробелы

Пробел, табуляция, перевод строки

#### Комментарии

Блочный, строчный, javadoc

#### Лексемы

### Виды лексем

#### Идентификаторы

Имена имеют пакеты, классы, интерфейсы, поля, методы, аргументы и локальные переменные

#### Ключевые слова

abstract double int strictfp

boolean else interface super

break extends long switch

byte final native synchronized

case finally new this

catch float package throw

char for private throws

class goto protected transient

const if public try

continue implements return void

default import short volatile

do instanceof static while

#### Литералы

##### Целочисленные литералы

##### Дробные литералы

##### Логические литералы

##### Символьные литералы

##### Строковые литералы

##### Null-литерал

##### Разделители

( ) { } [ ] ; , . ... @ ::

##### Операторы

= > < ! ~ ? : ->

== >= <= != && || ++ --

+ - \* / & | ^ % << >> >>>

+= -= \*= /= &= |= ^= %= <<= >>= >>>=

##### Пример программы

### Дополнение. Работа с операторами

#### Операторы присваивания и сравнения

#### Арифметические операции

#### Логические операторы

#### Битовые операции

## Типы данных

### Введение

### Переменные

три базовые характеристики:

имя;

тип;

значение.

Объявление, инициализация, final.

### Примитивные и ссылочные типы данных

#### Целочисленные типы

|  |  |  |
| --- | --- | --- |
| **Название типа** | **Длина (байты)** | **Область значений** |
| byte | 1 | -128 .. 127 |
| short | 2 | -32.768 .. 32.767 |
| int | 4 | -2.147.483.648 .. 2.147.483.647 |
| long | 8 | -9.223.372.036.854.775.808 .. 9.223.372.036.854.775.807 (примерно 1019) |
| char | 2 | '\u0000' .. '\uffff', или 0 .. 65.535 |

Над целочисленными аргументами можно производить следующие операции:

* операции сравнения (возвращают булево значение)
  + <, <=, >, >=
  + ==, !=
* числовые операции (возвращают числовое значение)
  + унарные операции + и -
  + арифметические операции +, -, \*, /, %
  + операции инкремента и декремента (в префиксной и постфиксной форме): ++ и --
  + операции битового сдвига <<, >>, >>>
  + битовые операции ~, &, |, ^
* оператор с условием ?:
* оператор приведения типов
* оператор конкатенации со строкой +

вычисления проводятся только с 32-х и 64-х битной точностью

#### Дробные типы

Название типа Длина (байты) Область значений

float 4 3.40282347e+38f ; 1.40239846e-45f

double 8 1.79769313486231570e+308 ; 4.94065645841246544e-324

Специальные значения:

* положительная и отрицательная бесконечности (positive/negative infinity);
* значение "не число", Not-a-Number, сокращенно NaN ;
* положительный и отрицательный нули.

#### Булев тип

### Ссылочные типы

#### Объекты и правила работы с ними

обращение к полям и методам объекта

оператор instanceof (возвращает булево значение)

операции сравнения == и != (возвращают булево значение)

оператор приведения типов

оператор с условием ?:

оператор конкатенации со строкой +

#### Класс Object

##### getClass()

##### equals()

##### hashCode()

##### toString()

##### finalize()

#### Класс Class

## Модуль компиляции и его структура

### Введение

### Имена

#### Простые и составные имена. Элементы

#### Имена и идентификаторы

#### Область видимости (введение)

### Пакеты

Пространство имен

#### Элементы пакета

#### Платформенная поддержка пакетов

classpath

#### Модуль компиляции

Cостоит из трех частей:

1. объявление пакета ;
2. import -выражения;
3. объявления верхнего уровня.

##### Объявление пакета

##### Импорт-выражения

Алгоритм приведения простого имени к полному:

1. сначала просматриваются выражения, импортирующие типы;
2. затем другие типы, объявленные в текущем пакете, в том числе в текущем модуле компиляции;
3. наконец, просматриваются выражения, импортирующие пакеты.

#### Объявление верхнего уровня

#### Уникальность имен пакетов

### Область видимости имен

#### Затеняющее объявление (Shadowing)

#### Заслоняющее объявление (Obscuring)

### Соглашения по именованию

### Модификаторы доступа

#### Предназначение модификаторов доступа

#### Разграничение доступа в Java

### Объявление классов

#### Заголовок класса

#### Тело класса

#### Объявление полей

#### Объявление методов

Сигнатура методов

#### Объявление конструкторов

#### Инициализаторы

### Дополнительные свойства классов

#### Метод main

#### Параметры методов

#### Перегруженные методы

## Преобразование типов

### Введение

### Виды приведений

#### Тождественное преобразование

#### Преобразование примитивных типов (расширение и сужение)

#### Преобразование ссылочных типов (расширение и сужение)

#### Преобразование к строке

#### Запрещенные преобразования

### Применение приведений

#### Присвоение значений

#### Вызов метода

#### Явное приведение

#### Оператор конкатенации строк

#### Числовое расширение

##### Унарное числовое расширение

##### Бинарное числовое расширение

### Тип переменной и тип ее значения

## Массивы

### Массивы как тип данных в Java

#### Объявление массивов

#### Инициализация массивов

#### Многомерные массивы

#### Класс массива

### Преобразование типов для массивов

#### Ошибка ArrayStoreException

#### Переменные типа массив и их значения

### Клонирование

#### Клонирование массивов

# Материалы для подготовки:

1. Спецификация языка Java
2. Курс по Java на intuit

<http://www.intuit.ru/studies/courses/16/16/info>

лекции 1-9