Павел Степанов

Кафедра компьютерной математики и программирования СПб ГУАП

Язык программирования Java

Тема 2. Основные понятия

- Базовые типы данных
- Основные понятия языка
- Классы, объекты, и их составляющие
- Инкапсуляция, наследование и полиморфизм
- Перегрузка vs переопределение
- Класс Object

2.3 Типы данных

- Примитивные
  - Целочисленные (byte, short, int, long)
    - Чему равен x? int x = 10000000;
    - Целочисленная арифметика по умолчанию – типа int
  - Вещественные
    - Вещественная арифметика по умолчанию – типа double
  - boolean
    - If (a=b)
- Объектные
  - Object и его наследники
  - String
  - Врапперы простых типов
  - Массивы
  - Как правильно сравнивать объекты

2.4 String

- Immutable
  - StringBuilder и StringBuffer
- Пул строк
  - В чем разница между “a” и new String(“a”);
  - Метод intern()
  - “a”+”b” == “ab” или нет?
- Методы класса String
  - Метод substring
  - Метод split
  - Метод trim
  - Метод replace
  - Метод append
  - Методы toUpperCase и toLowerCase

2.5 Пул строк – как сделано?

```
String s = "a" + "b"; //        ldc           #2          // String ab
                      //        astore_1
String s1="a";        //        ldc           #3
                      //        astore_2
String s2="b";        //        ldc           #4
                      //        astore_3
String s3=s1+s2;      //        new           #5          // class java/lang/StringBuilder
                      //        dup
                      //        invokespecial #6          // Method java/lang/StringBuilder."<init>":()V
                      //        aload_2
                      //        invokevirtual #7          // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
                      //        aload_3
                      //        invokevirtual #7          // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
                      //        invokevirtual #8          // Method java/lang/StringBuilder.toString:()Ljava/lang/String;
                      //        astore        4
                      //        return
```

- литералы обрабатываются на этапе компиляции,
- операции над строками – через класс StringBuilder.
- String.intern –нативный метод

- Интересная статья - http://java-performance.info/string-intern-in-java-6-7-8/
  - Пул находится в куче, размер определяется параметром - XX:StringTableSize=N
  - Организован в виде хэш-таблицы

2.6 Врапперы простых типов

- Позволяют достигать единообразия в использовании примитивов и объектных типов
- immutable
- Автоматические boxing и unboxing
  - Для массивов не работают!
- Пулы чисел
- Разница между parseXXX и valueOf

2.7 Пул чисел – как сделано?

```
Integer x = 100;     // bipush        100
                     // invokestatic  #9                  // Method java/lang/Integer.valueOf:(I)Ljava/lang/Integer;
                     // astore_1
                     // return
```

```
public static Integer valueOf(int i) {
    assert IntegerCache.high >= 127;
    if (i >= IntegerCache.low && i <= IntegerCache.high)
        return IntegerCache.cache[i + (-IntegerCache.low)];
    return new Integer(i);
}
```

2.8 Массивы

- Массив – это и объект, и область памяти
  - У него есть атрибуты и методы, например, length
- Инициализация
  - int[] arr1=new int[10];
  - int[][] arr2=new int[15][20];
  - int[][] arr3=new int[15][];
- Если у двух массивов базовые классы являются сабтипом и супертипом, то в том же отношении находятся и массивы
- Вариант long[] arr=new int[10] не работает ни в какой форме

2.9 Основные понятия

- ООП
- Пакеты
- Классы
- Объекты
- Методы классов
- Атрибуты классов

2.10 ООП

- Инкапсуляция
  - Классы
  - Объекты
  - Атрибуты
  - Методы
- Наследование
  - Интерфейс
- Полиморфизм
  - Переопределение и перегрузка

2.11 Пакеты

- Связь пакета, пути к файлу и CLASSPATH/-classpath
- Пакет по умолчанию
- www.mycompany.com -> com.mycompany
- Пакетная видимость
- Компиляция и запуск файлов, лежащих в пакетах

2.12 Классы

- Имя класса
- Видимость класса
  - Модификатор public
  - Модификатор private
  - Видимость по умолчанию
- Атрибуты класса
- Методы класса
- Конструктор
- секции инициализации
- Специальные модификаторы
  - Модификатор abstract
  - Модификатор final

2.13 Объекты

- Оператор new
  - Связь с конструктором и секциями инициализации
- Деструктора – нет!
  - Метод finalize – не деструктор
- Динамические и статические атрибуты и методы
- Значение атрибутов по умолчанию

2.14 Атрибуты

- Динамические атрибуты
- Статические атрибуты
- Значение по умолчанию
- Видимость атрибутов
  - Модификатор public
  - Модификатор private
  - Модификатор protected
  - Видимость по умолчанию
- Специальные модификаторы
  - Модификатор volatile
  - Модификатор transient
  - Модификатор final

2.15 Методы

- Динамические методы
- Передача параметров
- Статические методы
- Специальные модификаторы
  - Модификатор synchronized
  - Модификатор strictfp – IEEE 754
  - Модификатор abstract
  - Модификатор native
  - Модификатор final

2.16 Инициализация

- Конструктор
  - Конструктор по умолчанию
  - Несколько конструкторов
  - Вызов одного конструктора из другого
    - Вызов this()
    - Вызов super()
  - Паттерн Singleton
  - Конструктор копирования
- Секция динамической инициализации
- Секция статической инициализации

2.17 Порядок инициализации

- Статические инициализаторы
  - Суперкласса
  - Субкласса
- Динамическая инициализация суперкласса
  - Секции инициализации
  - Конструктор
- Динамическая инициализация субкласса
  - Секции инициализации
  - Конструктор

2.18 Специальные вопросы

- Почему методически неверно, когда конструктор кидает исключения?

2.17 Наследование

- Наследование от классов
  - Субкласс и суперкласс
  - Наследование атрибутов
  - Object
  - Нет множественного наследования
- Видимость при наследовании
- Конструкторы и инициализация при наследовании
  - Наследование без конструктора по умолчанию
- Совместимость при присвоении
- Оператор instance of
- Методы и классы, объявленные как final
- Приведение типов

2.18 Интерфейсы

- Абстрактные классы
- Интерфейсы
  - мотивация
  - Наследование от интерфейсов
    - Наследование абстрактных классов
    - Наследование неабстрактных классов
  - Расширение интерфейсов
  - Изменения в JDK 8
    - Default method

2.19 default method (JDK 8+)

```
public interface A { default void foo(){
    System.out.println("Calling A.foo()"); } }

public interface B { default void foo(){
    System.out.println("Calling B.foo()"); } }

public class C implements A { }

public class D implements A, B {
    public void foo(){ A.super.foo(); }
}
```

2.20 Специальные интерфейсы

- Serializable
- Cloneable
- AutoCloseable (JDK 7+)
- Iterable
- Runnable

2.21 Перегрузка

- В классе могут быть методы с одинаковым названием, но разной сигнатурой
  - Сигнатура - типы и количество параметров
  - Возвращаемое значение не входит в сигнатуру

2.22 Правила перегрузки

- Подбор метода осуществляется в следующем порядке:
  - Без использования boxing/unboxing и оператора …
  - С использованием boxing/unboxing, но без оператора …
  - С использованием boxing/unboxing и оператора …

2.23 Переопределение

- Возвращаемое значение – то же самое, или субкласс (для объектов)
- Типы и количество параметров должны совпадать
- Метод должен генерировать те же контролируемые исключения
- Чтобы не промахнуться - @Override
- Переменные не подвержены полиморфизму

2.24 Класс Object

- Предок всех классов
  - Конструктор
  - Метод finalize
  - Методы wait, notify, notifyAll
  - Методы equals и hashCode
  - Метод toString
  - Метод clone
  - Метод getClass

2.25 Q & A