НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Типы данных и управляющие структуры Java

Евгений Беркунский, НУК

eugeny.berkunsky@gmail.com

http://berkut.homelinux.com

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Что такое тип данных?

Тип данных:

- Спектр значений
- Набор допустимых операций

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Программа

- Примитивные и ссылочные типы
- Размеры данных
- Преобразование и приведение типов
- Управляющие конструкции
- Массивы
- Строки
- Основы ООП

Примитивные и ссылочные типы

- Язык Java является объектно-ориентированным, но существуют типы данных (простые/примитивные), не являющиеся объектами.
  - Фактор производительности.
- Простые типы делятся на 4 группы:
  - целые: int, byte, short, long
  - числа с плавающей точкой: float, double
  - символы: char
  - логические: boolean
- Введение в синтаксис языка классов позволяет создавать свои типы, получившие название ссылочных

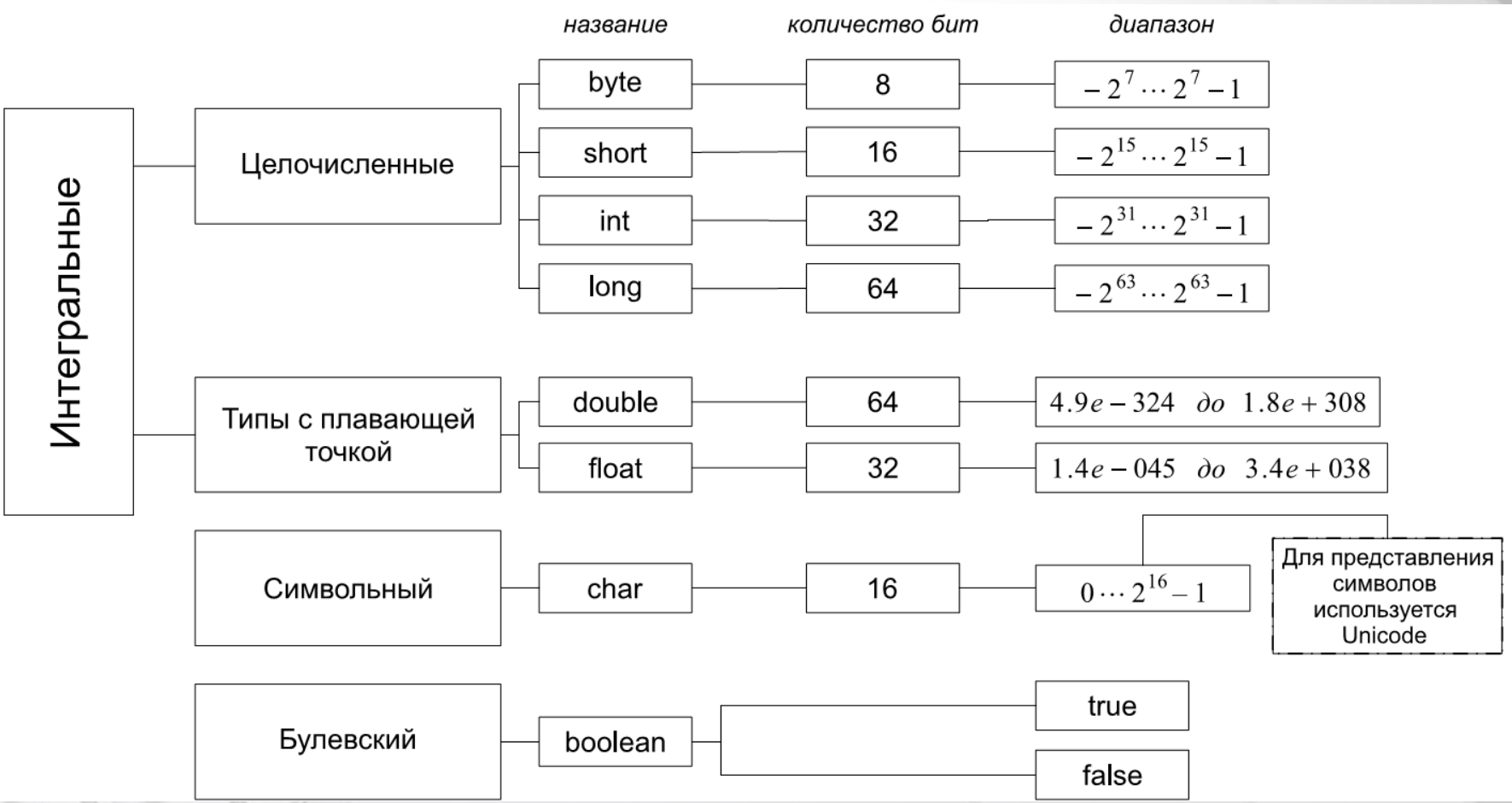
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Примитивные типы

Примитивный тип	Размер (бит)	Минимальное значение	Максимальное значение	Класс - оболочка
boolean	-	-	-	Boolean
char	16	‘\u0000’	‘\uFFFF’	Character
byte	8	-128	127	Byte
short	16	-2^{15}	$2^{15}-1$	Short
int	32	-2^{31}	$2^{31}-1$	Integer
long	64	-2^{63}	$2^{63}-1$	Long
float	32	IEEE754	IEEE754	Float
double	64	IEEE754	IEEE754	Double
void	-	-	-	Void

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Примитивные типы



Значения по умолчанию

- Неинициализированная явно переменная (член класса) примитивного типа принимает значение в момент создания:

Примитивный тип	Значение по умолчанию
boolean	false
char	‘\u0000’ (null)
byte	(byte)0
short	(short)0
int	0
long	0L
float	0.0f
double	0.0 (0.0d)

Неинициализированная локальная переменная – ошибка компиляции.

Классы-оболочки

- Иногда переменные примитивных типов должны быть использованы к качестве объектов.
- Для каждого примитивного типа Java в пакете java.lang существует класс-оболочка: Long, Integer, Float,…
- Класс-оболочка – немодифицируемый класс

```
char symbol = 'x';
Character symbol = new Character(symbol);
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Числа с высокой точностью

- Предназначены для проведения расчетов без потери точности.
- Не имеют примитивных аналогов.
- BigInteger – для представления длинных целых чисел.
- BigDecimal – для представления больших чисел с плавающей точкой.

```
BigInteger one = BigInteger.ONE;
BigInteger thousand = new BigInteger(1000);
```

Приведение типов

- Приведение – это изменение типа (автоматическое или явное).
- Явное приведение позволяет :
- сделать тип преобразования более точным.
- форсировать преобразование, если оно не может быть выполнено автоматически.

```
int x = 200;
long y = (long)x;
long value = (long)200; // необязательно, т.к. компилятор
                        // делает это автоматически

long value = 1000L;
int value2 = (int)value; //обязательно.
// Иногда это единственный способ сделать код компилируемым
```

Приведение ссылочных типов

- Java позволяет проводить приведение от любого примитивного типа к любому примитивному типу, за исключением boolean.
- Для классов действуют несколько другие правила приведения.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Ссылочные типы данных

```
String s; //создание ссылки
s = “Hello”; //присвоение значения
```

Для большинства классов используется оператор new:

```
String s1 = new String(“World”);
```

Программист имеет возможность создавать свои ссылочные типы

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Константы

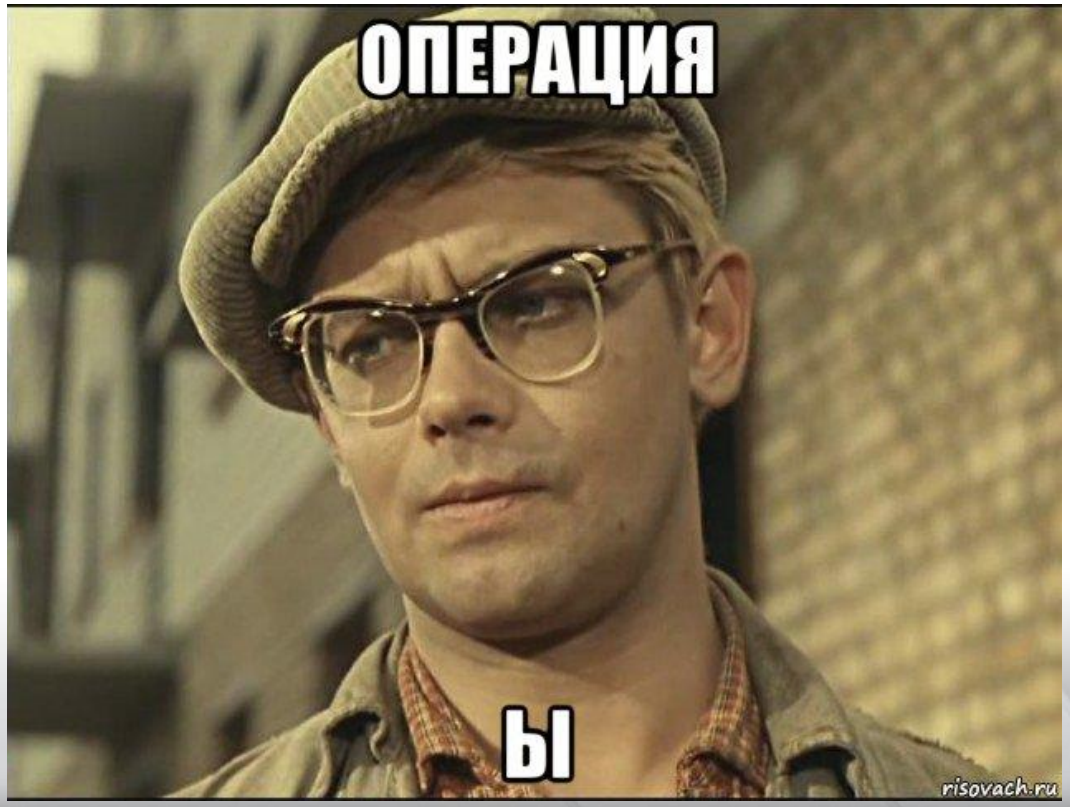
Переменные с неизменяемыми значениями называются константами

```
final int MAXIMUM_SPEED = 250;
```

Операции

Выделяют четыре типа операций:

- арифметические,
- поразрядные,
- отношений,
- логические.



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Арифметические операции

- Операнды должны иметь числовой тип.
- Можно использовать операции на типе char.

Знак	Операция
+	Сложение
-	Вычитание
*	Умножение
/	Деление
%	Остаток от деления
++	Инкремент
--	Декремент

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Поразрядные (битовые) операции

- Действуют на индивидуальные биты.
- Применяются для числовых операндов

Знак	Операция
~	Отрицание
&	И – AND
|	ИЛИ – OR
^	Исключающее ИЛИ – XOR
>>	Сдвиг вправо
<<	Сдвиг влево
>>>	Сдвиг вправо с заполнением старшего бита нулем

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Операции отношений

- Определяют отношения, которые один операнд имеет к другому

Знак	Операция
==	Равно
!=	Не равно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Логические операции

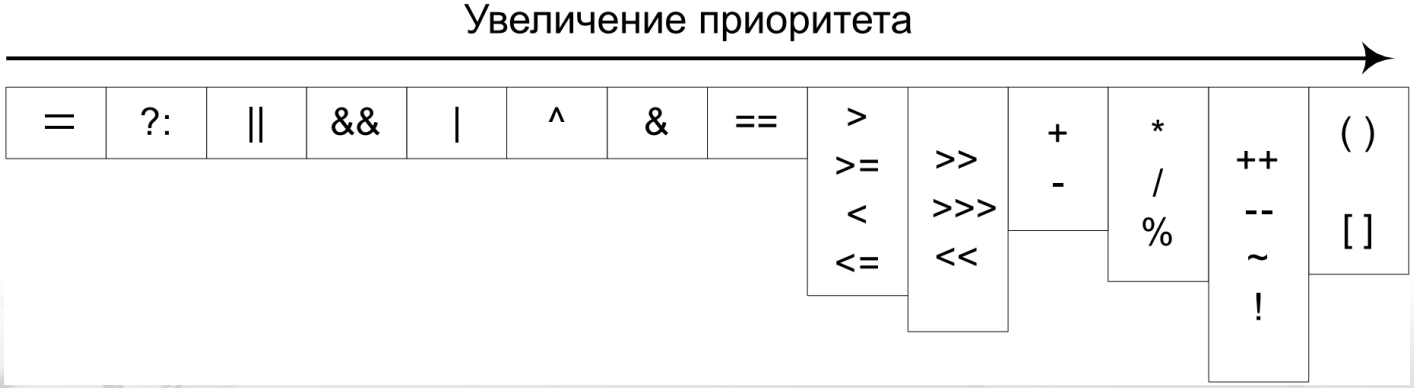
- Работают только с типом boolean.

Знак	Операция
!	Отрицание
&&	И – AND
||	ИЛИ – OR
^	Исключающее ИЛИ – XOR
==	Равно
!=	Не равно
? :	Троичная условная операция

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Приоритеты операций

- Операторы Java практически совпадают с операторами C++ и имеют та-кой же приоритет, как приведен на рисунке
- Операторы работают с базовыми типами, для которых они определены, и объектами классов-оболочек над базовыми типами.
- Кроме этого операторы «+» и «+=» производят также действия по конкатенации операндов типа String.
- Логические операторы «==», «!=» и оператор присваивания «=» применимы к операндам любого объектного и базового типов, а также литералам.



Размещение сущностей

Переменная может быть размещена в одном из двух хранилищ:

- Стек – расположен в памяти VM и имеет поддержку в виде указателя стека.
- Куча (heap) – область памяти VM, в которой хранятся все ссылочные типы.
- Переменной простого типа выделяется память в стеке.
- Ссылки располагаются в куче.

Операторы управления

Оператор условного перехода **if** имеет следующий синтаксис:

```
if (boolean_значение) { /* операторы */ } // 1
else { /* операторы */} // 2
```

- Если выражение boolean_значение принимает значение true, то выполняется группа операторов 1, иначе — группа операторов 2.
- Оператор **else** может отсутствовать, в этом случае операторы, расположенные после окончания оператора **if** (строка 2), выполняются вне зависимости от значения булевского выражения оператора **if**

Операторы управления

Оператор множественного выбора switch:

```
switch(value) {
    case val1: /* операторы */
               break; /* не обязателен*/
    ...
    case valN: /* операторы */
               break;
    default: /* операторы */
}
```

- При совпадении значения value со значением val1 выполняется следующий за ним вариант.
- Затем, если отсутствует оператор break, выполняются подряд все блоки операторов до тех пор, пока не встретится оператор break.
- Если значение value не совпадает ни с одним из значений в case, то выполняется блок default.
- Значения val1,…, valN должны быть константами и могут иметь значения типа int, byte, short, char или enum. В Java7 в этот список включен и тип String

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Операторы управления

В Java существует четыре вида циклов.

Сначала первые три:

```
while (boolean_значение) { /* операторы */}
// цикл с предусловием

do { /* операторы */} while (boolean_значение);
// цикл с постусловием

for (выражение_1; boolean_значение; выражение_3) {
    /* операторы */
} // цикл с параметрами
```

В версии 5.0 введен еще один цикл, упрощающий доступ к массивам и коллекциям:

```
for (ТипДанных имя : имяОбъекта) { /* операторы */}
// цикл полного перебора
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Массивы

- Для хранения нескольких однотипных значений используется ссылочный тип – массив:

```
//примитивный тип, размер массива задан явно
int[] price = new int[10];

//неявное задание размера
int[] rooms = new int[]{1, 2, 3};

//содержит ссылочные переменные
Item[] items = new Item[10];

Item[] undefinedItems = new Item[]{
     new Item(1),
     new Item(2),
     new Item(3)
};
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Доступ к элементам массива

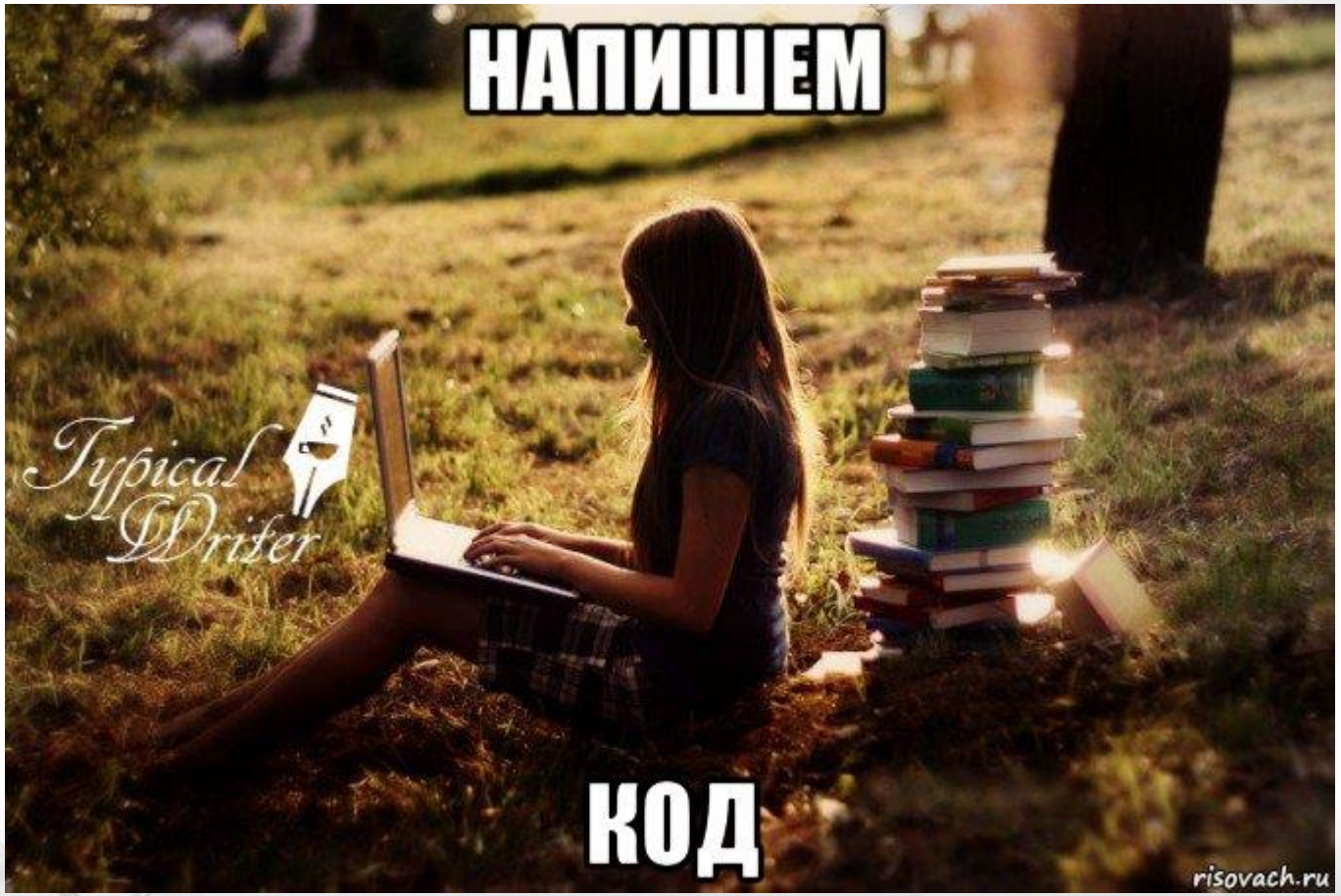
- Доступ осуществляется по индексу.
- Размер хранится в немодифицируемом поле массива length

```
for (int i = 0; i < undefinedItems.length; i++) {
     // должен быть представим в виде строки -
     // переопределен метод toString()

     System.out.println(undefinedItems[i]);
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрация



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Многомерный массив

- Является массивом массивов.
- Многомерный массив концептуально представляет собой многомерную матрицу

```
int[][] twoDim = new int[4][5];
```

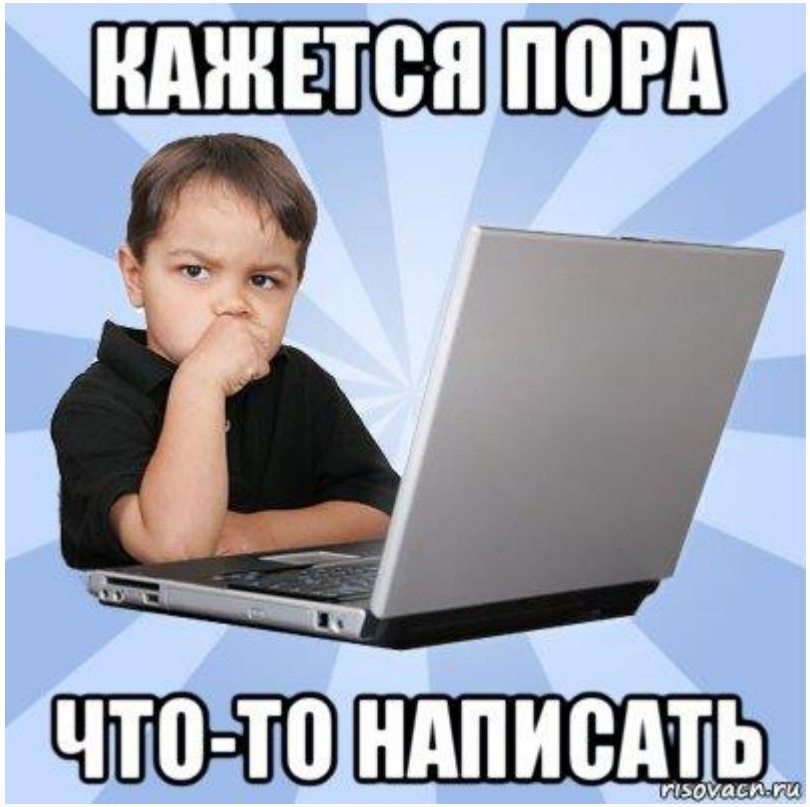
Массив массивов

Каждый из массивов может иметь отличную от других длину

```
int[][] twoDim = new int[4][];
twoDim[0] = new int[10];
twoDim[1] = new int[20];
twoDim[2] = new int[30];
twoDim[3] = new int[100];
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрация



Строки

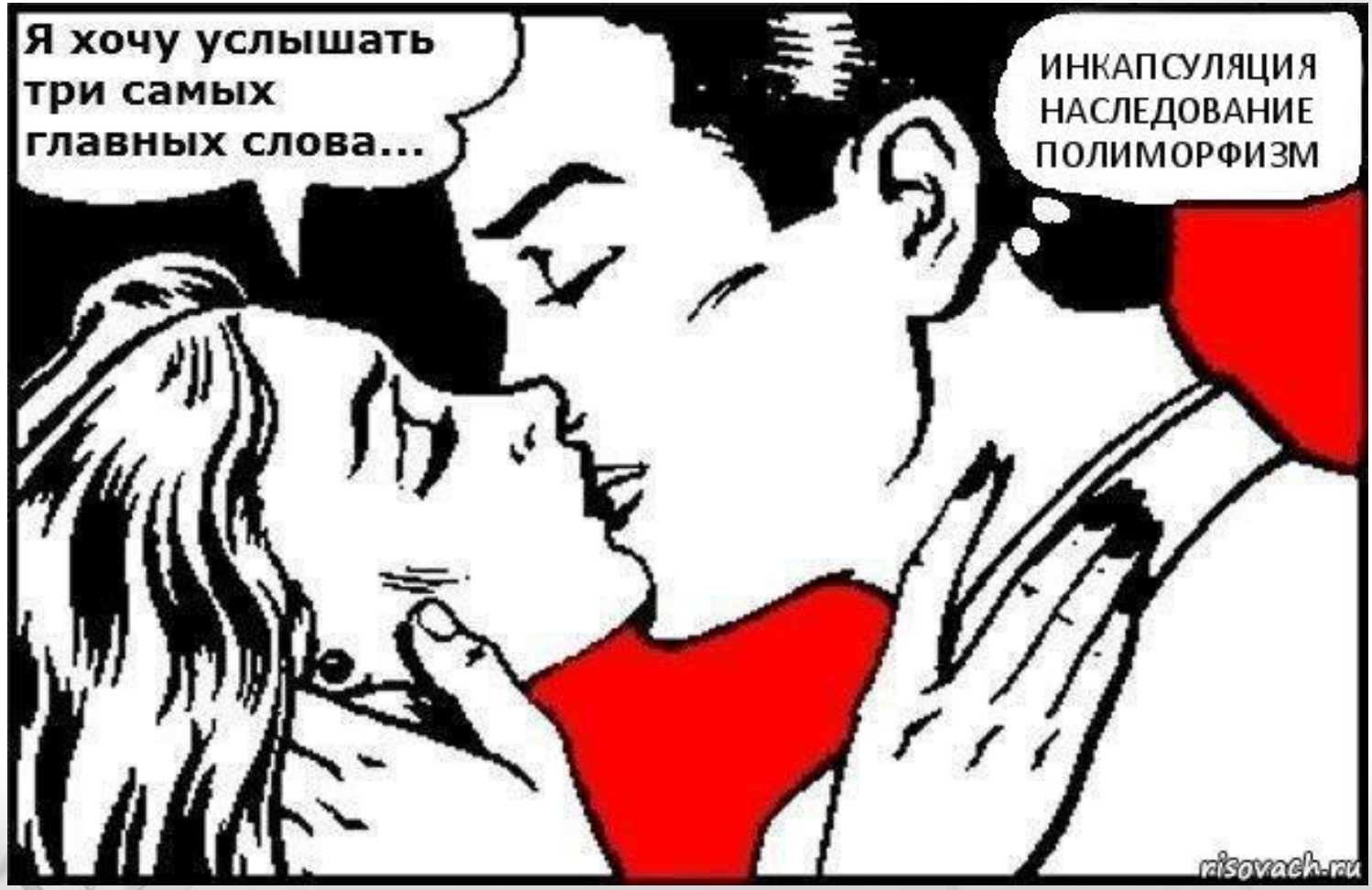
- Класс java.lang.String представляет собой хранилище символов и функциональность для работы с ними.
- Строка имеет фиксированную длину и не завершается специальным символом.
- Возможности:
  - обращение к символу по его номеру,
  - поиск,
  - выделение подстроки,
  - изменение регистра,
  - и т.п.
- Объект класса String – неизменяем.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрация

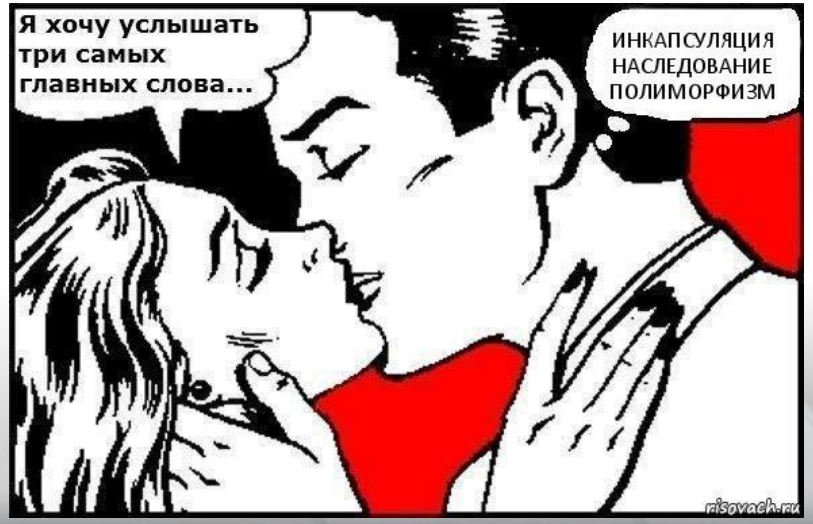
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Основы ООП



Основы ООП

- *Инкапсуляция* (сокрытие данных) - объединение в одной сущности данных и методов работы с ними.
- *Наследование* – возможность класса-наследника приобретать признаки класса-предка.
- *Полиморфизм* – способность наследников по-другому реализовывать возможности предков. Объект способен проявлять признаки своего предка.



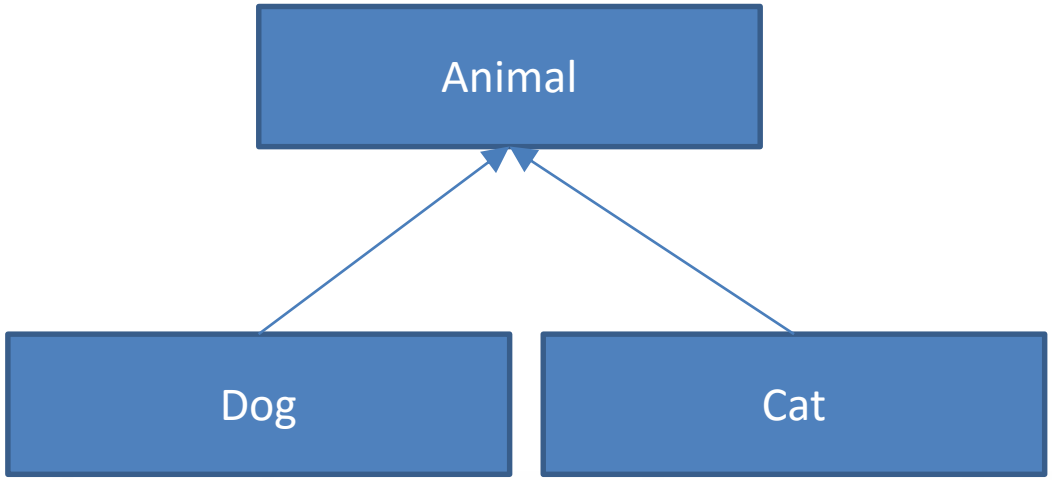
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Инкапсуляция

- Состояние объекта определяется значениями его полей.
- Сокрытие данных осуществляется с помощью установки модификаторов, влияющих на видимость членов класса.
- В языке Java используется несколько уровней сокрытия.

Наследование

- Наследование - это такое отношение между классами, когда один класс повторяет структуру и поведение другого класса (одиночное наследование) или других классов (множественное наследование – отсутствует в Java).



Полиморфизм

- Экземпляр класса может выступать как экземпляр любого класса-предка данного класса.

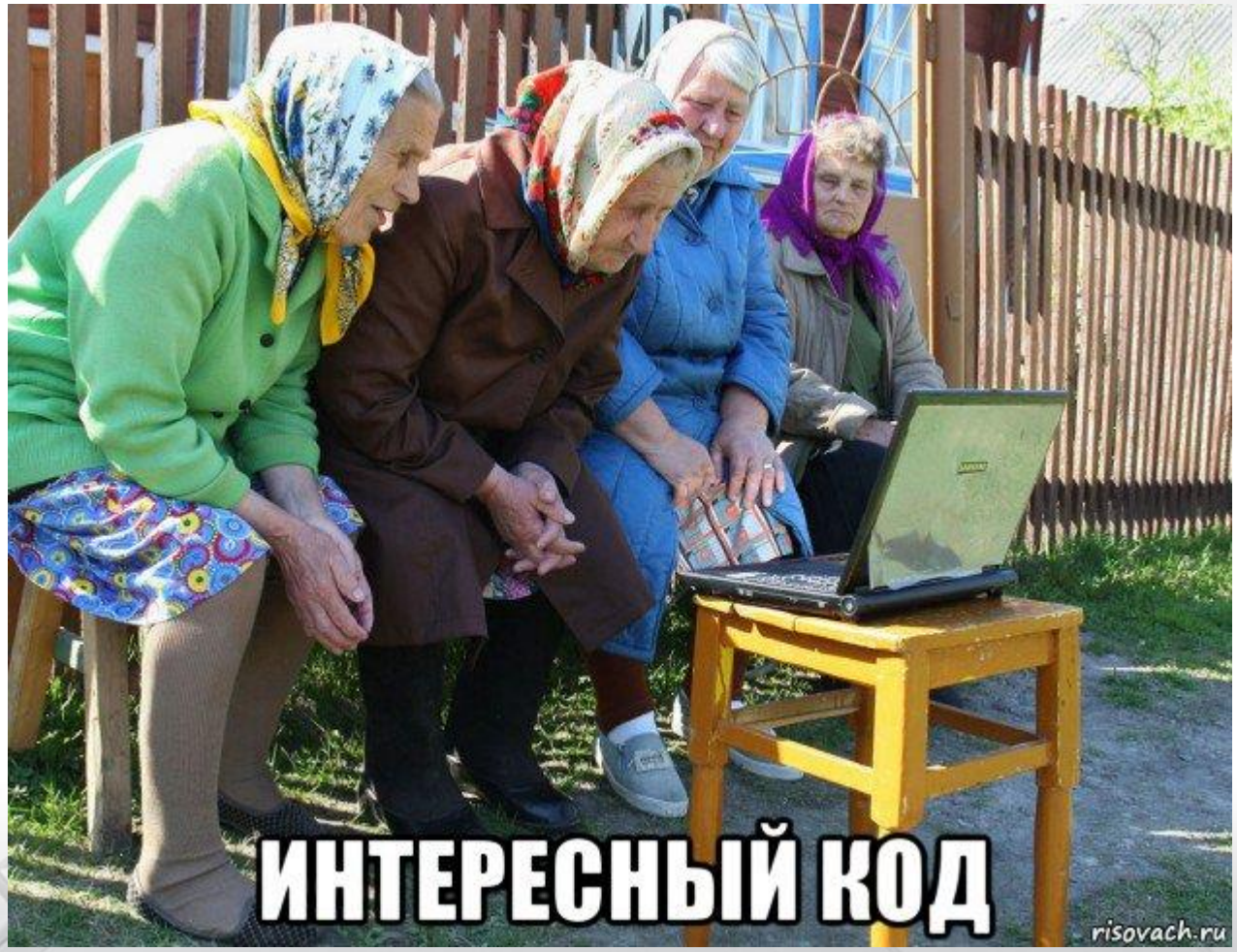
```
class Animal {
   void talk(){…};
}
class Dog extends Animal {
    void talk() { System.out.println("Woof!"); }
}
class Cat extends Animal {
    void talk() { System.out.println("Meow!"); }
}
```

Экземпляр класса Dog или Cat одновременно является экземпляром класса Animal:

```
Animal myDog = new Dog();
Animal myCat = new Cat();
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрация




QUESTIONS
& ANSWERS

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Типы данных и управляющие структуры Java

Евгений Беркунский, НУК

eugeny.berkunsky@gmail.com

http://berkut.homelinux.com