Programming

Numbers, strings and arrays

4th lecture

Overview

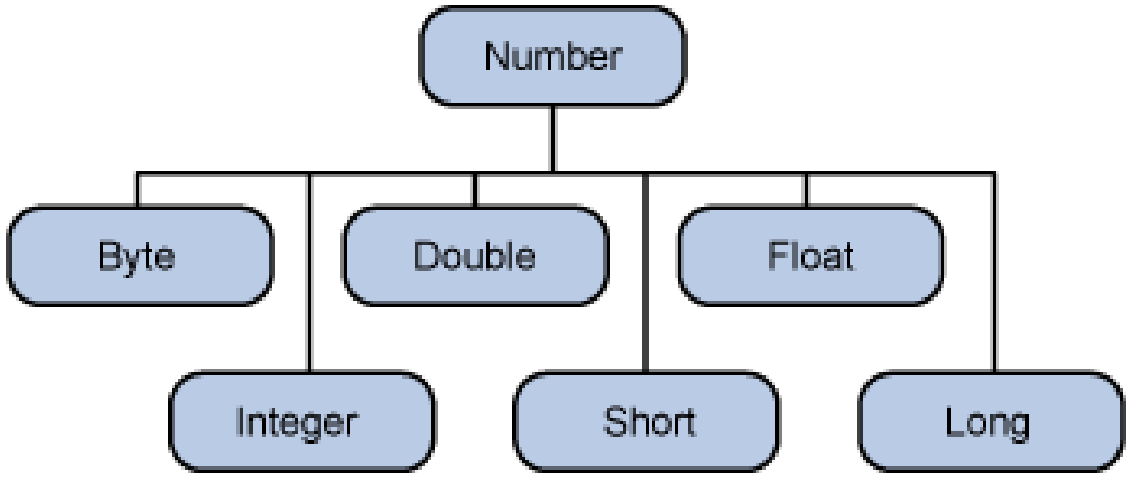
- Numbers' presentation:
  - Basic data types
  - Classes
- Stringd
- Arrays

Basic data types

- We already covered the subject:
  - int, byte, short, long
  - double, float
  - char
  - ~~boolean~~

Class `java.lang.Number`

- Extended from **Number**:
  - Byte
  - Short
  - Integer
  - Long
  - Float
  - Double



- Classes instead of basic data types ➔ **ovijanje** (*wrapping*)

Why classes?

- 3 reasens for class usage instead of basyc data types
  - When the parameter of a methos is an object
  - The need for MIN_VALUE in MAX_VALUE
  - The usage od methods (**metod)** for translation/transformation between basic data types
- Example:

```
Integer x, y;
x = 12;
y = 15;
System.out.println(x+y);
```

Methods pf the “number” classes

- Methods of the Number class and subclasses:

http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Number.html

- Why are they useful?
  - … homework …

Characters – **`char`** and **`Character`**

- Examples:

```
char ch = 'a';
char uniChar = '\u039A';
  // Unicode reppresentation of the Greek omega
char[] charArray ={ 'a', 'b', 'c', 'd', 'e' };
  // array of characters
Character ch = new Character('a');
Character ch = 'a';
  // basic type 'a' is wrapped into and object of the
  type Character
Character test(Character c) {...}
  // object of the type Character is parameter of the
  method and the type of the result of the method
char c = test('x');
  // wrapped and unwrapped
```

Methods of the class **`Character`**

- **`boolean isLetter(char ch)`**
- **`boolean isDigit(char ch)`**
- **`boolean isWhiteSpace(char ch)`**
- **`boolean isUpperCase(char ch)`**
- **`boolean isLowerCase(char ch)`**
- **`char toUpperCase(char ch)`**
- **`char toLowerCase(char ch)`**
- **`toString(char ch)`**

- What are special characters? (**`\t`**,**`\b`**,**`\n`**,**`\r`**,**`\f`**,**`\'`**,**`\"`**,**`\\`**)

Strings - definition

- Examples of the defintions of strings (*String*):

```
String greeting = "Hello world!";

char[] helloArray = { 'h', 'e', 'l', 'l', 'o', '.'};

String helloString = new String(helloArray);

System.out.println(helloString);
```

Strings – length, concatenation

- How do we establish the length of a string?

```
String palindrome = "Dot saw I was Tod";
int len = palindrome.length();
```

- How to concatenate strings?

```
string1.concat(string2);
"My name is ".concat("Rumplestiltskin");
"Hello," + " world" + "!"
```

Strings ++

- What can we do with strings?
  - Convert them into numbers and vice-versa
  - Process characters in strings:
    - Find characters, substrings
    - Change case
    - Change characters, substrings, …

http://java.sun.com/docs/books/tutorial/java/data/manipstrings.html
http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html

Arrays

- Declaration, initialization and access:

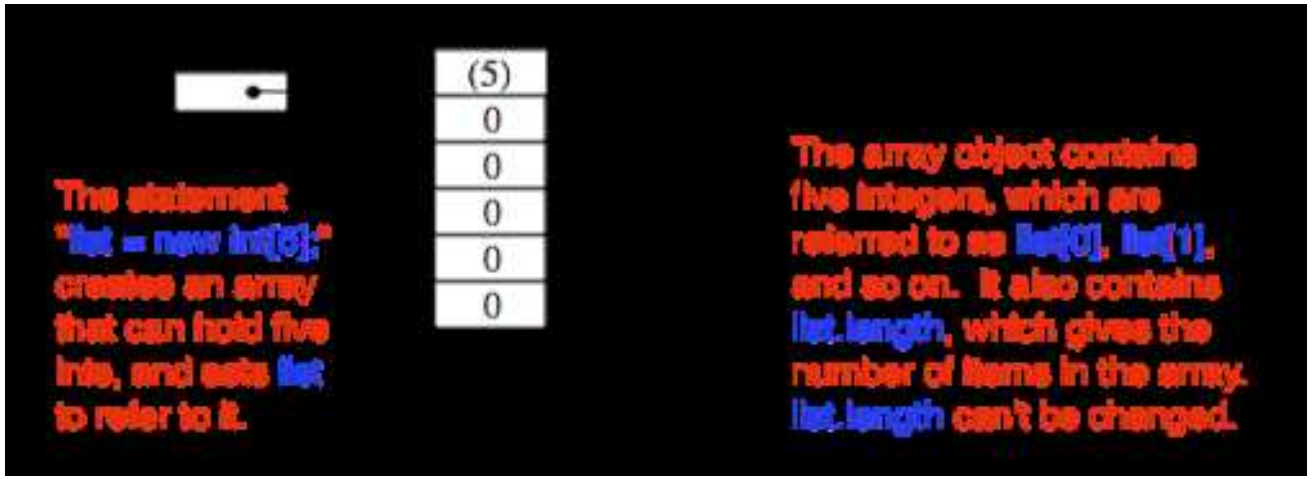
```
class ArrayDemo {
     public static void main(String[] args) {
          int[] anArray; // declaration of an array of integer numbers
          anArray = new int[10]; // memory allocation for 10 integers
          anArray[0] = 100; // initialization of the 1. element
          anArray[1] = 200; // initialization of the 2. element
          anArray[2] = 300; // ...
        ...
```

- …all in one:

```
int[] anArray = {100, 200, 300, 400, 500, 600};
```

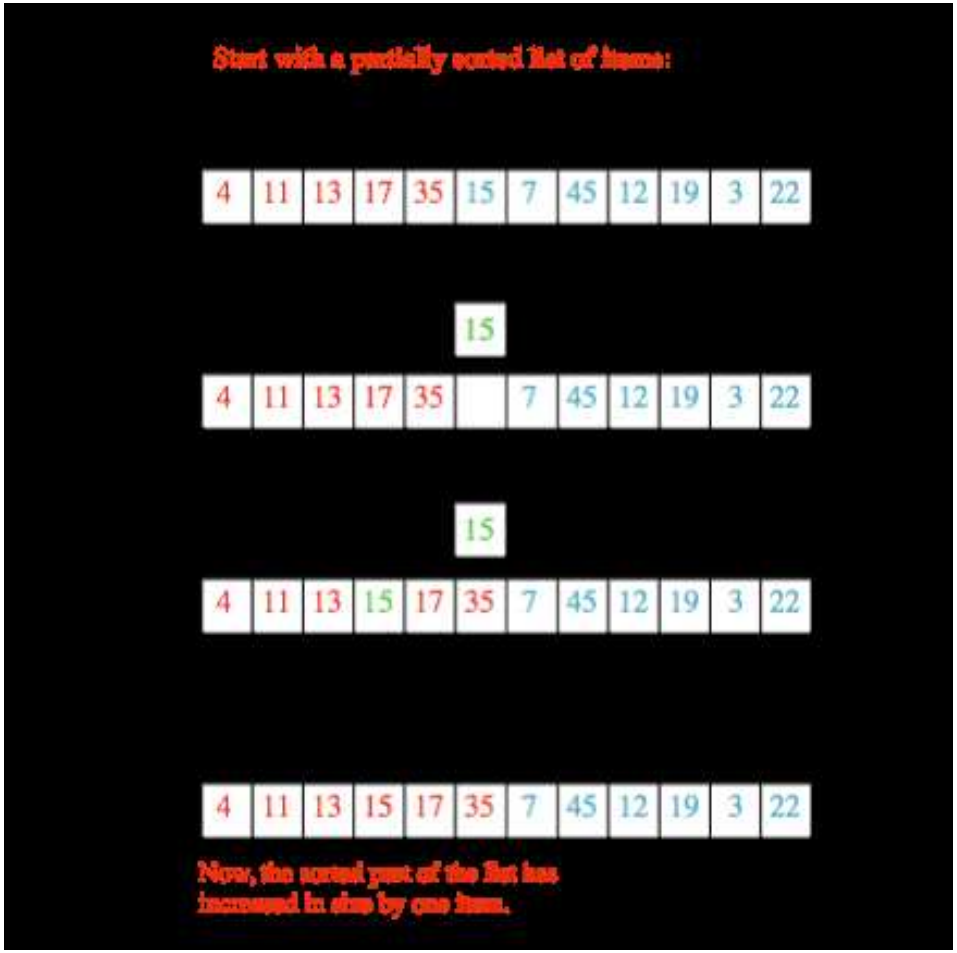
More about arrays … (1)

- In the memory:



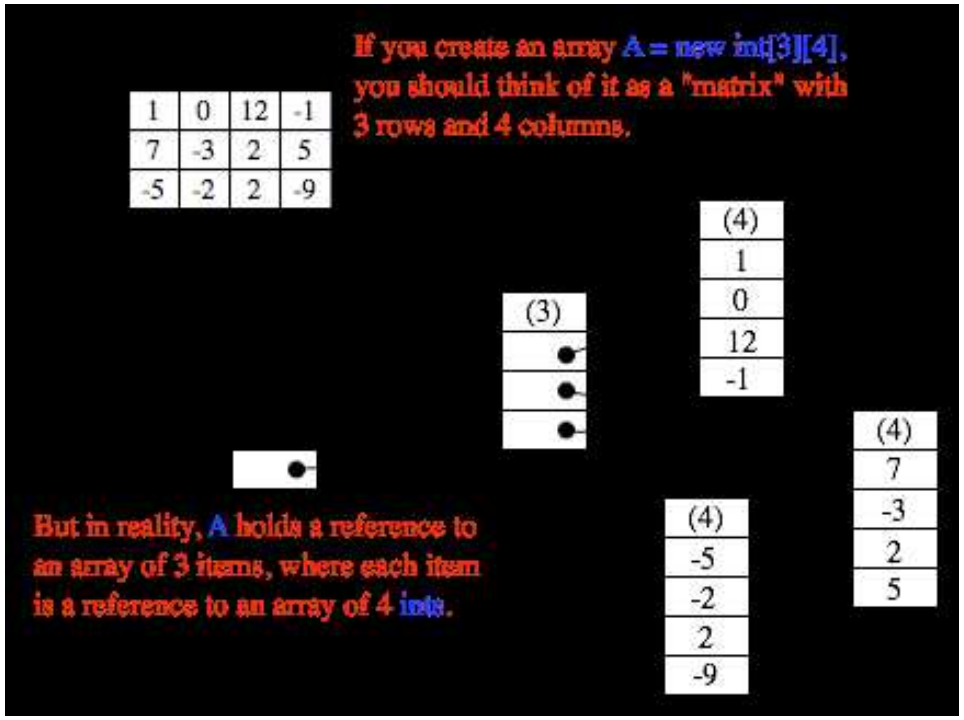
More about arrays … (2)

- Arrays for search and sorting



More about arrays … (3)

- Multi-dimensional arrays



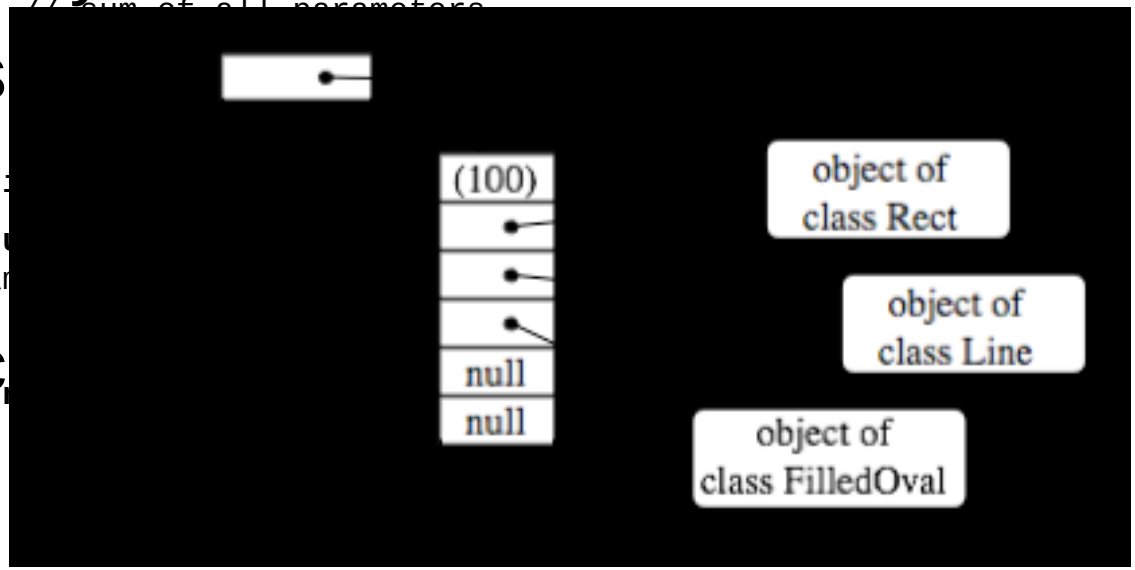

```
int[][]  A  =  {  {  1,  0, 12, -1 },
                  {  7, -3,  2,  5 },
                  { -5, -2,  2, -9 }
               };


int[][]  A  =  new int[3][4];
for (int row = 0;  row < 3;  row++) {
   for (int column = 0;  column < 4;
         column++) {
      A[row][column] = 0;
   }
}
```

More about arrays … (4)

- Dynamic arrays:
  - Variable-length arrays
  - "Half-empty" arrays
  - Special objects:
    - **ArrayList**
    - **Vector**
  - Let us write a c

```
public static double average( double[] numbers ) {
    double sum;
    double average;
    sum = 0;
    for (int i = 0; i
        sum = sum + nu
        // add a paran
    }
    average = sum / n
    return average;
}
```



Summary

- What have we smo izvedeli?
  - About “number classes”, wrapping of basic data types, …
  - About characters and and strings
  - About arrays and possible usages

Resources

- http://java.sun.com/docs/books/tutorial/java/data/index.html
- http://java.sun.com/docs/books/tutorial/java/nutsandbolts/arrays.html
- http://math.hws.edu/javanotes/c7/index.html

Homework

1. Remember the homework from 123 minutes before?