Computer Programming

Standard Library

The `java.lang` package

Willy Picard
Department of Information Technology
The Poznan University of Economics
<picard@kti.ae.poznan.pl>

Agenda

- Lecture Goal(s)
- Refreshments
- Overview of the `java.lang` package
- The `Object` class
- The `String` and `StringBuffer` classes
- The `System` and `Runtime` classes
- Conclusion

Lecture Goal(s)

Lectures Overview

Fundamental Concepts

- 1: Introduction
- 2: Basic data structures & Statements
- 3: Object-oriented programming I
- 4: Object-oriented programming II
- 5: Object-oriented programming III
- 6: Complex data structures
- 7: Threads and Exception handling

Lectures Overview

Java

- 8: Summarizing Example
- 9: Standard library
- 10: GUI – AWT
- 11: GUI – Swing
- 12: IO programming
- 13: Network programming
- 14: Java archives and JavaBeans
- 15: Conclusions

Today's Goal

To provide programming knowledge about the standard library in the **`java.lang`** package

LECTURE GOALS

Refreshments

The API Specification

- Documentation
  - Packages
  - Interfaces
  - Classes
  - Inheritance
  - Attributes
  - Methods

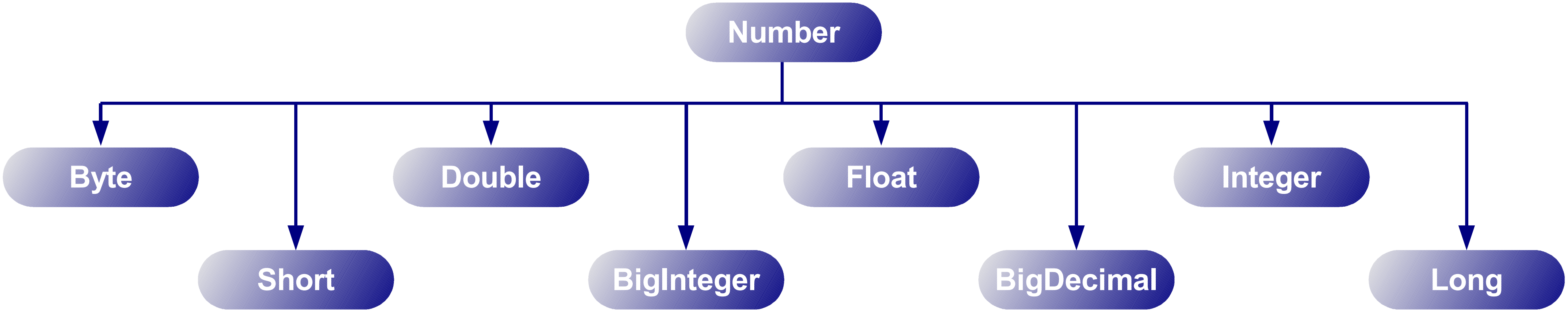
Overview of the `java.lang` package

From the API: `java.lang`

Provides classes that are fundamental to the design of the Java programming language

Primitive Wrappers

- Number
  - Byte
  - Double
  - Float
  - Integer
  - Long
  - Short
  - BigDecimal
  - BigInteger
- Boolean
- Character
- Void



Why Primitive Wrappers?

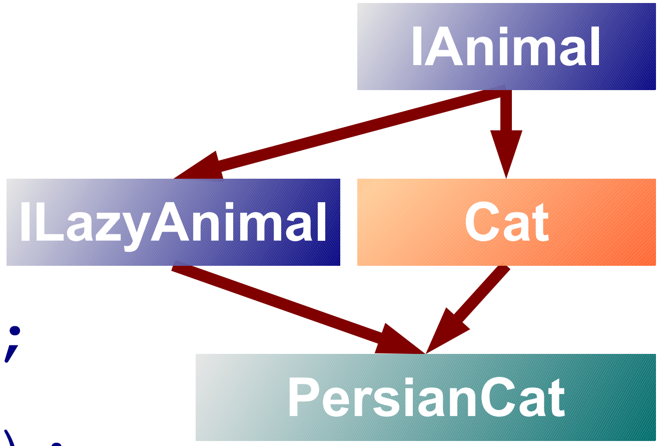
- Incorrect example
  - List myNumbers = new ArrayList();
  - int a = 1;
  - myNumbers.add(a);
- Correct example
  - List myNumbers = new ArrayList();
  - Integer a = new Integer(1);
  - myNumbers.add(a);

Other Classes

- The `Object` class
- Support for strings of characters
  - The `String` class
  - The `StringBuffer` class
- Interaction with the JVM
  - The `System` class
  - The `Runtime` class

The `Object` Class

One Object, Various Forms in Java



- ~~`IAnimal cat = new IAnimal();`~~
- `IAnimal cat = new Cat();`
- `IAnimal cat = new PersianCat();`
- ~~`IAnimal cat = new ILazyAnimal();`~~
- `Cat cat = new Cat();`
- `Cat cat = new PersianCat();`
- `ILazyAnimal cat = new PersianCat();`
- `PersianCat cat = new PersianCat();`

The `Object` Class

- `Object cat = new Cat();`
- Every class extends `Object` class
- The `Object` class provides
  - A set of methods
  - An empty constructor

The `Object` Methods

- The `toString()` method
  - Called in `System.out.println(myObj);`
- The `equals()` method
  - Object comparison
- The `hashCode()` method
  - Hash function
  - Must be implemented if the `equals()` method is redefined
  - Equal objects must have equal hashCode

The `hashCode` Method

- `int result = 17;`
- For each significant field compute c
  - Boolean → `c = ( f ? 0 : 1)`
  - byte, char, short, int → `c = (int) f;`
  - Long → `c = (int) (f ^ (f >>> 32));`
  - Float → `c= Float.floatToIntBits(f)`
  - Double →
    ```
    d = Double.doubleToLongBits(f);
    c = (int) (d ^ (d >>>32));
    ```
  - Object → `c = ((f == null)? 0:f.hashCode());`
  - Combine c and f: `result = 37*result+c;`

Example

Complete Cat

String **Class**
StringBuffer **Class**

The `String` Class

- Character strings
- Constants
- Strings as arrays
  - `length()`
  - `charAt()`
- Example

```
String name=”Willy Picard”;
int length = name.length(); // 12
char c = name.charAt(0);    // 'W'
```

The String Class

- String manipulation
  - indexOf()
  - substring()
- Example

```
String name = “Willy-Picard”;
int index = name.substring(“Pic”);
String first = name.substring(0,index);//Willy-
String last = name.substring(index);   //Picard
```

STRING

The `StringBuffer` Class

- Modifiable character strings
- The append() method

```
String space = “ “;
StringBuffer buff = new StringBuffer();
buff.append(“Willy”);
buff.append(space);
buff.append(“Picard”);
```

- The toString() method

```
String name = buff.toString();
```

Example

String Example

STRING

System **Class**
Runtime **Class**

The `System` Class

- A set of useful fields and methods
- No constructors
- Only **static** fields and methods
- Three fields
  - in: standard input
  - out: standard output
  - err: standard error output

Example

Standard Streams Example

The `System` Properties

- A set of properties for the working environment
- (Key, Value) pairs
- Information about
  - The JVM
  - The operating system
  - The user
  - Separators
    - Line, File, Path

Example

System Properties
Example

Miscellaneous Methods

- Array copy
- Current time
  - `currentTimeMillis()`
  - Number of milliseconds since January 1, 1970 UTC
- Garbage Collector
  - `gc()`
  - Force garbage collection

Example

Time Example

S
Y
S
T
E
M

The `Runtime` Class

- Use the `Runtime.getRuntime()` to get an instance
- Information about memory
  - freeMemory()
  - maxMemory()
  - totalMemory();
- Execution of external commands
- Beware!
  - **not cross-platform**

Example

Runtime Example

Conclusion

The Java Standard Library

- Check the API!!!
- Fundamental Classes

See you next week