Packages

Class is a module

- Interface
  - Operations and non-private methods
  - Non-private constants

not private ≠ public

- Implementation
  - Operations and private methods
  - Private attributes
  - Method bodies

- Contracts
  - Pre and post-conditions of operations and methods

- User manual
  - Class documentation
  - Interface documentation

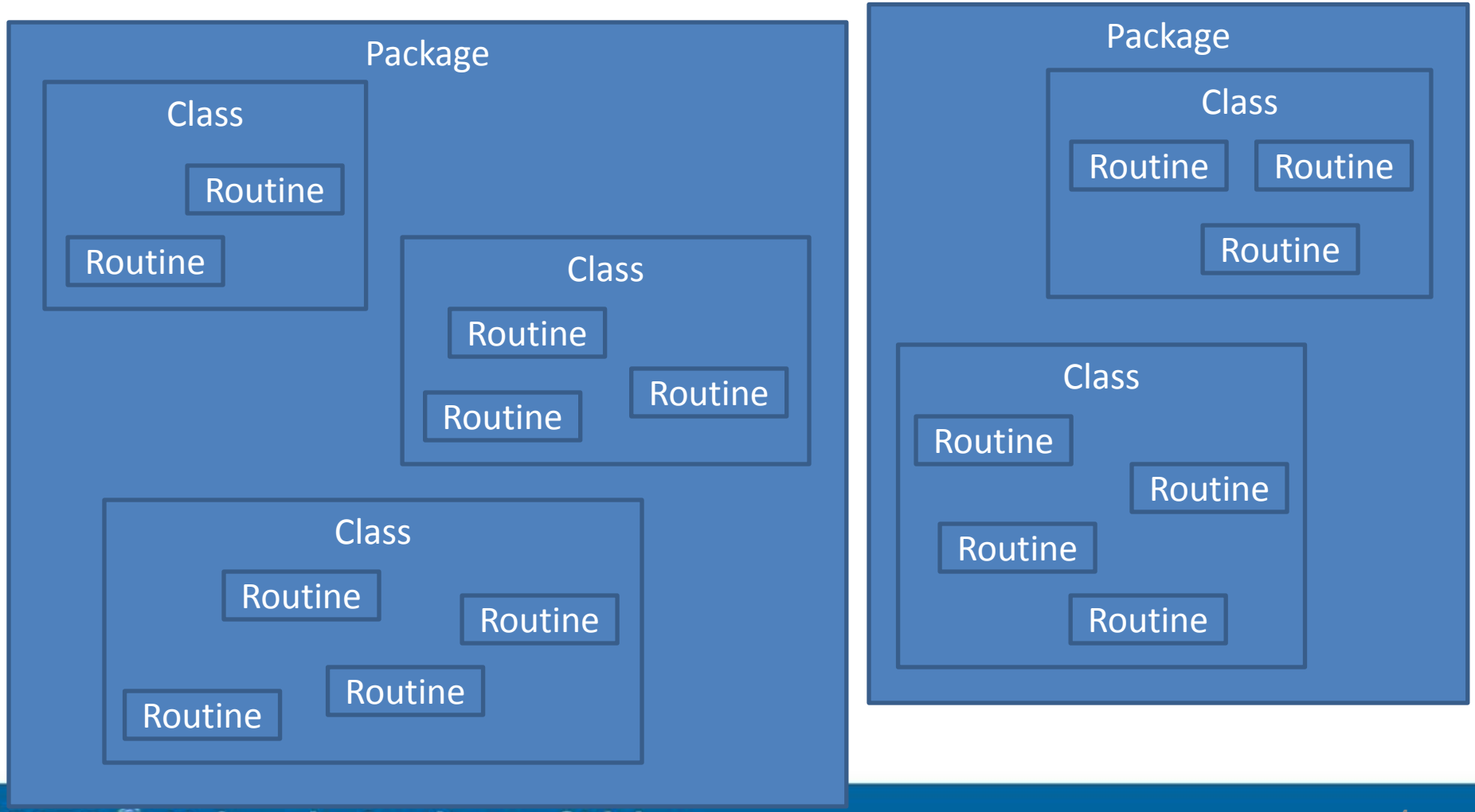
Access categories

- Members can be:
  - `private` – only accessible to member of the same class

    (In Java) Access is possible between different objects of the same class!
  - *package-private* (no qualifier) – also accessible to members in the same package
  - `protected` – also accessible to members of derived classes (… next lessons)
  - `public` – universal access

More accessible

Modules in Java



Packages

- Sets of classes strongly related
- Examples
  - `java.util`
  - `org.junit`
- Name conventions
  - Small caps
  - No word separation
  - Abreviations and acronyms acceptable
  - First elements are the inverted DNS name (e.g., `pt.iscte`)
  - Remaning elements may reference strutural units (e.g., `pt.iscte.dcti.poo`)

Packages as modules

- Interface
  - Public classes
  - Non private members of public classes
- Implemention
  - Beyond class implementation …
  - … all package-private classes

Packages: hierarchy

- java
  - lang
  - util
- org
  - junit
  - omg
- pt
  - iscte
    - dcti
      - ip
        - » games
      - poo

Game.java

```
package pt.iscte.dcti.poo

class Game {
    …
}
```

Player.java

```
package pt.iscte.dcti.poo

class Player {
    …
}
```

- Open hierarchy
  - No isolated declaration
  - Each `.java` file declares its package

Packages: hierarchy

- Relevant for naming
  - Logical organization (in folders)
  - Lower name collision

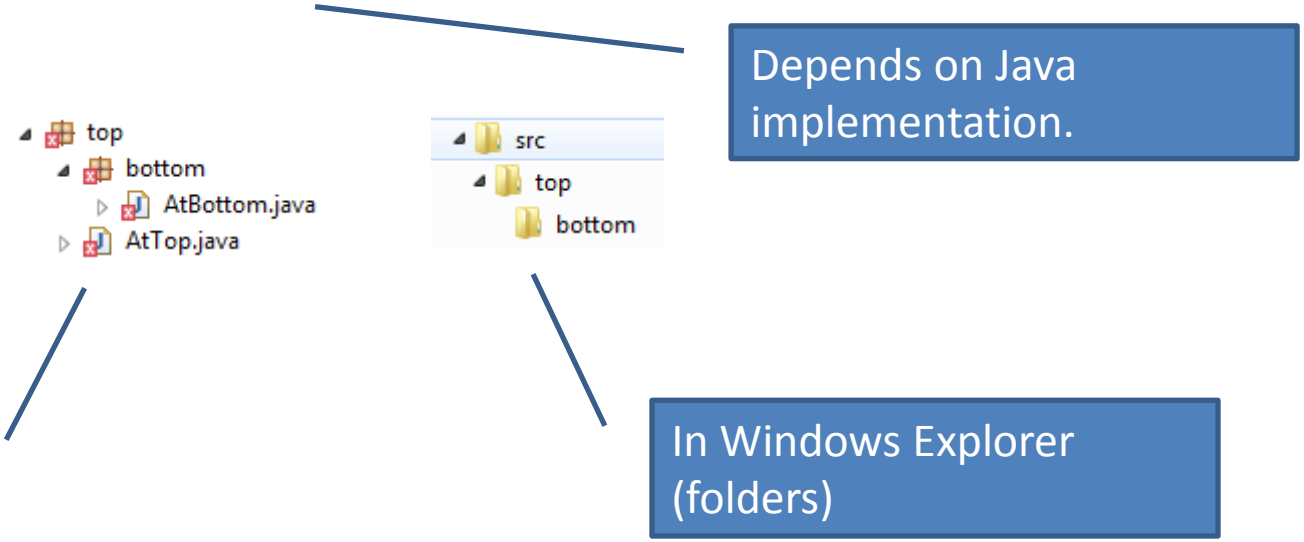
- Irrelevant for access
  - Package and subpackage are independent
  - Members of subpackage are not package members
  - Members of package are not subpackage members

Files

- One public class per file
- Must have same name (class and file)
- Any number of *package-private* classes per file, but …
- Good practices: One class per file!

Folders

- Usually package hierarchy corresponds to folders



Class vs. Package Class

Class

- Template for object creation

```
public class Calculator {
   private int value;

   public Calculator() {
      ...
   }

   public int value() {
      ...
   }

   public void clear() {
      ...
   }
   ...
}
```

Package class

- Set of related methods

```
public class CharUtils {
   public static char nextLetter(char c) {
      ...
   }

   public static char previousLetter(char c) {
      ...
   }

   public static int distance(char x, char y) {
      ...
   }
   ...
}
```

Class methods and attributes (`static`)

- Key-word `static` means class member (as opposed to instance member)
- To use a class method must import class
- static methods have no access to instance attributes

Class method (`static`)

Use of class method

```
char c = CharUtils.nextLetter('a');
```

Different from other methods

```
Calculator myCalculator = new Calculator();

myCalculator.clear();
```

More information / References

- Y. Daniel Liang, "Introduction to Java Programming" 7th Ed. Prentice-Hall, 2010.

Summary

- Packages