UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica etsinf

Unit 5. Control structures: Selection

Introduction to Computer Science and Computer Programming
Introducción a la Informática y la Programación (IIP)

Year 2017/2018

Departamento de Sistemas Informáticos y Computación

Contents

Contents

Introduction and motivation

- As seen in Unit 3, most real problems need to take decisions depending on the situation, and choose among different instruction subsequences
- Selection control structures allow to obtain these features
- In a method, decisions are usually taken depending on:
  - Input data (parameters)
  - State of the object (attributes)
  - Intermediate data
- This unit presents details on the different decision mechanisms in Java

Introduction and motivation

Conditional instructions can be simple or nested

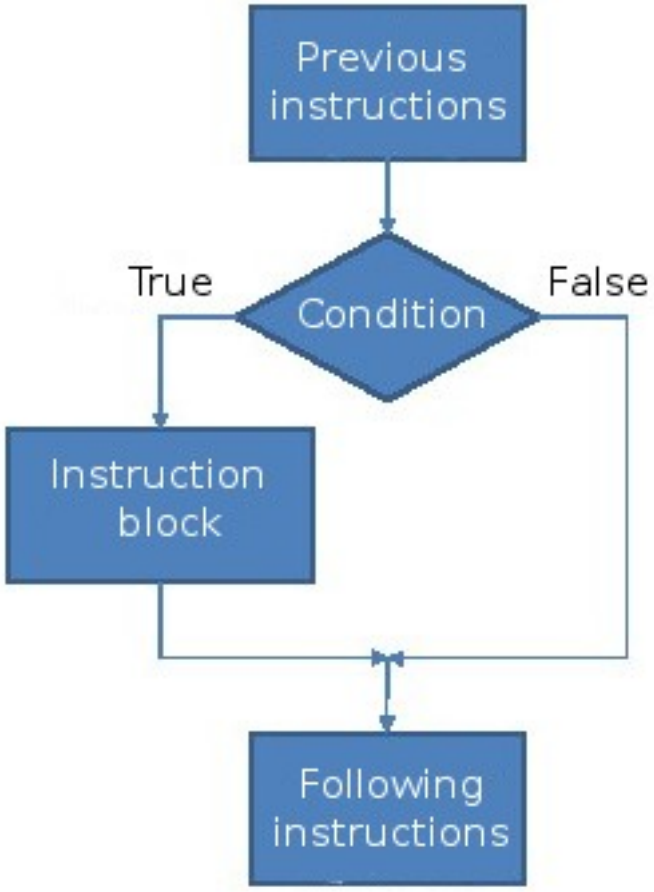
```
if (temp>100)
   System.out.println("Temperature greater than 100");
else System.out.println("Temperature lower than or equal to 100");
System.out.println("Continuing here...");
//.....
```

```
if (temp>100)
   System.out.println("Greater than 100");
else if (temp>50)
   System.out.println("Greater than 50, lower than or equal to 100");
else System.out.println("Lower than or equal to 50");
System.out.println("Continuing here");
//.....
```

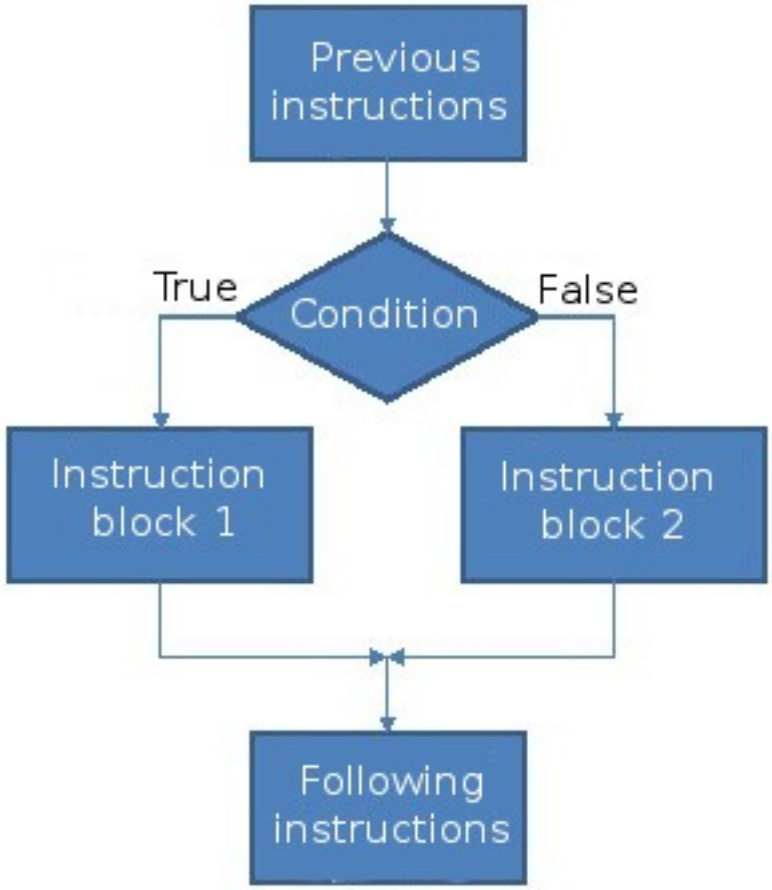
Contents

Simple and double decision: `if`, `if-else`

`if` conditional in Java:



`if-else` conditional in Java:



Simple and double decision: `if`, `if-else`

Example: `Circle` class, `setRadius` method:

```
/** updates the radius of the Circle to newRadius
    only when newRadius is positive or zero. */
public void setRadius(double newRadius) {
  if (newRadius >= 0)
    radius = newRadius;
}
```

Example: medical data, depending on sex, you ask if the person is pregnant

```
Human h = new Human();
// ...
System.out.print("How old are you? ");
// ...
if (h.getSex()=='W') {
  System.out.print("Are you pregnant? ");
  // ...
}
System.out.print("Are you allergic to any medicine? ");
// ...
```

Simple and double decision: `if`, `if-else`

Example: depending on the temperature, switch on or off the air conditioning:

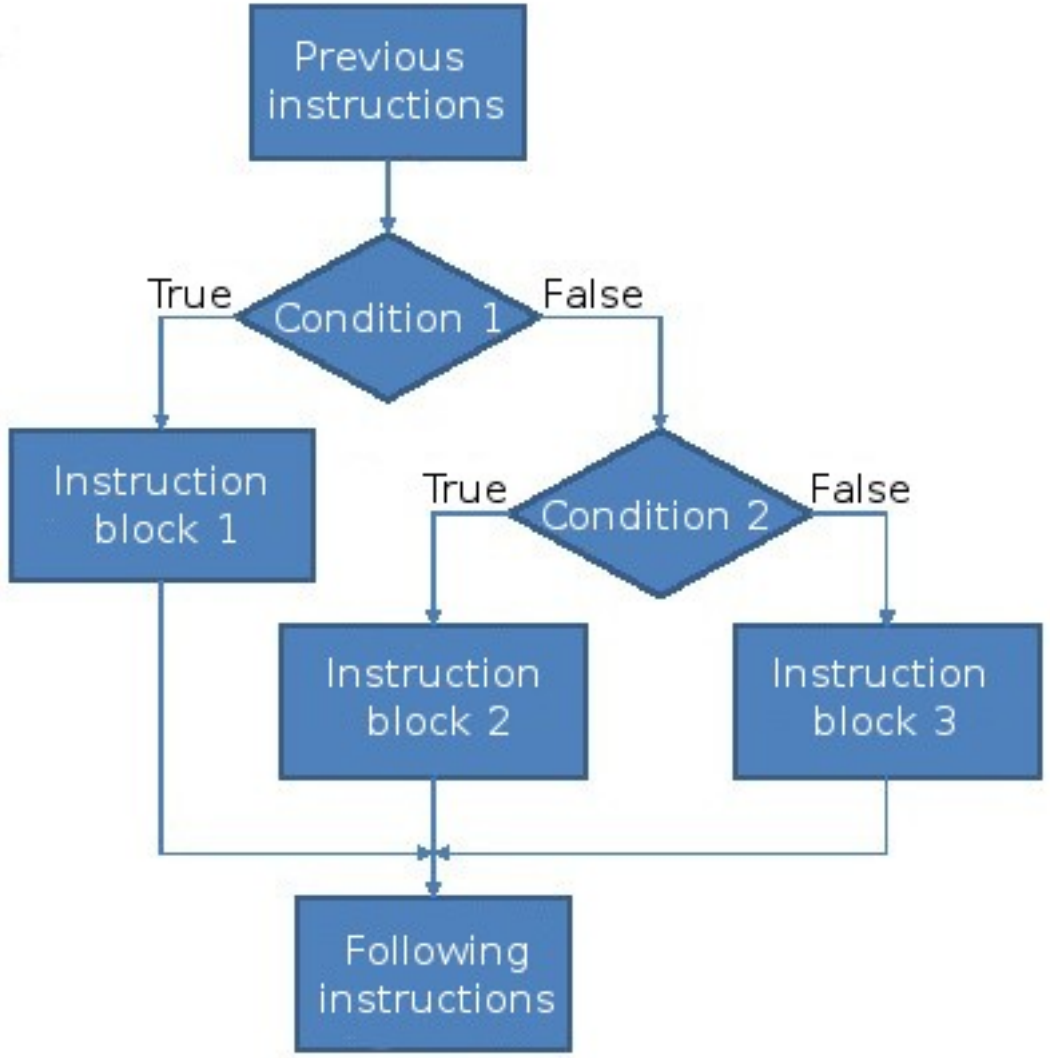
```
Room r = new Room();
Aircon a = new Aircon();
// ...
if (r.getTemp()>26.0)
  a.switchOn();
else
  a.switchOff();
// ...
```

Example: depending the sunlight intensity, regulate the lights intensity

```
SunlightDetec sl = new SunlightDetec();
LightBulb l1 = new LightBulb();
LightBulb l2 = new LightBulb();
// ...
if (sl.getIntensity()<100) {
  l1.incrIntens();
  l2.incrIntens();
}
else {
  l1.decrIntens();
  l2.decrIntens();
}
// ...
```

Simple and double decision: `if`, `if-else`

`if` and `if-else` instructions can be ***nested***



Simple and double decision: `if`, `if-else`

Example: a manager for traffic fines

```
// ...
if (vehic.getSpeed()>road.getMaxSpeed()) {
  System.out.println("Max speed exceeded");
  if (vehic.getSpeed()>1.2*road.getMaxSpeed())
    vehic.createSpecialFine();
  else
    vehic.createNormalFine();
}
else {
  if (vehic.getSpeed()<road.getMinSpeed())
    vehic.createNormalFine();
}
// ...
```

Simple and double decision: `if`, `if-else`

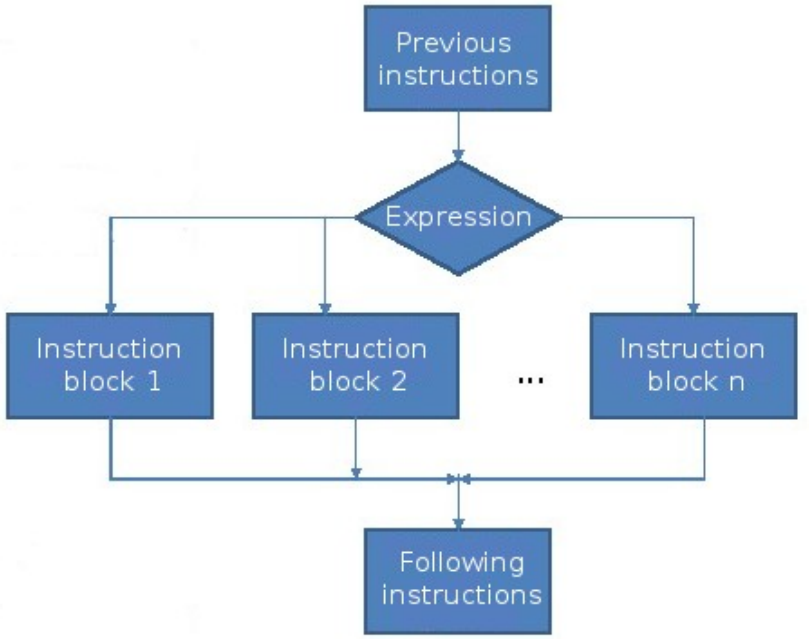
A complete example: validate a date (method `validate`)

```
public class Date {
  int dd, mm, yy;  // Day, month, year
  // ...
  public boolean leap() { return ( (yy%4)==0 && (yy%100)!=0 ) || (yy%400)==0; }

  public boolean validate() {
    int numdd=31;      // Number of days of the month

    if ((mm<=0) || (mm>12)) return false;  // Check month range
    // Determine day range
    if ((mm==4) || (mm==6) || (mm==9) || (mm==11)) numdd=30;
    else if (mm==2)
      if (leap()) numdd=29; else numdd=28;
    if ((dd<=0) || (dd>numdd)) return false;  // Check day range

    return true;
  }
  // ...
}
```

Contents

Multiple decision: `switch`

```
switch (expression) {
    case val_1:  [SC_1] [break;]
    case val_2:  [SC_2] [break;]
    ...
    ...
    case val_n:  [SC_n] [break;]
    [default:  [SC_{n+1}] [break;]]
}
```



- `expression` must be evaluated to any primitive type (except `float` or `double`) or to `String` (from Java 7.0 version)
- $\texttt{val}_1$, $\texttt{val}_2$, . . . , $\texttt{val}_n$ are literals of datatype compatible with `expression`
- SC_1, SC_2, . . . , SC_{n+1} are any sequence of instructions
- Components inside brackets (`[]`) are optional

Multiple decision: `switch`

Execution

1. Evaluate `expression`

2. Compare its value with all the alternatives in the `case`

3. If they are equal, execute all the instructions from that point **including code associated to other `case` labels** until:

   - A `break` is found
   - The `switch` block finishes

4. If no value is equal, execute instructions starting `default` label (if exists) till the end of the block

5. Continue with the instruction that follows the `switch` block

Multiple decision: `switch`

A complete example: return month name from month number

```
public class Date {
  int dd, mm, yy;  // Day, month, year
  // ...

  public String monthName() {
    String monthName;
    switch(mm) {
      case 1: monthName=new String("January"); break;
      case 2: monthName=new String("February"); break;
      case 3: monthName=new String("March"); break;
      case 4: monthName=new String("April"); break;
      // ...
      case 12: monthName=new String("December"); break;
      default: monthName=new String("Unknown");
    }
    return monthName;
  }
  // ...
}
```

Contents

The ternary conditional operator

Java, as C and C++, provides a ternary operator

```
boolexpr ?  expr1 :  expr2
```

- Behaviour similar to a conditional instruction
- `boolexpr` is a boolean expression
- `expr1` and `expr2` are any expression, but of the same datatype
- Execution:
  1. Evaluate `boolexpr`
  2. If `boolexpr` is `true`, evaluate whole expression to `expr1`
  3. If `boolexpr` is `false`, evaluate whole expression to `expr2`

The ternary conditional operator

Example: validate a date using the ternary operator

```
public class Date {
  int dd, mm, yy;  // Day, month, year
  // ...
  public boolean leap() { return ( (yy%4)==0 && (yy%100)!=0 ) || (yy%400)==0; }

  public boolean validate() {
    int numdd=31;      // Number of days of the month

    if ((mm<=0) || (mm>12)) return false;  // Check month range
    // Determine day range
    if ((mm==4) || (mm==6) || (mm==9) || (mm==11)) numdd=30;
    else if (mm==2) numdd=(leap())?29:28;   // Here ternary operator is used

    if ((dd<=0) || (dd>numdd)) return false;  // Check day range

    return true;
  }
  // ...
}
```