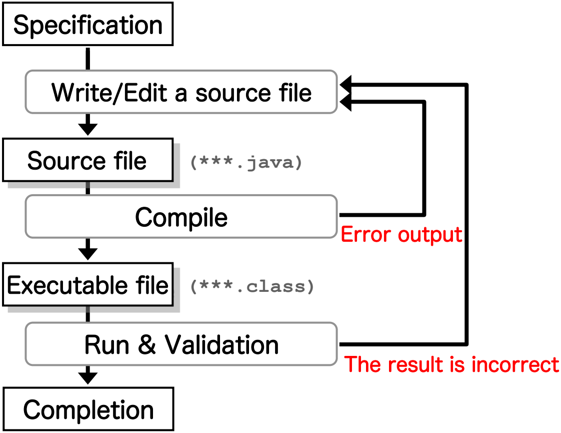
Java Programming

— Basics of Java Programming: Compile · Run —

Waseda University

Flow of programming

Flow of java program



Write/Edit a source file

Example

Output “Hello.” message on the screen.

⇓

Edit a source file

- Edit a source file by using a text editor.
- Emacs, gedit, vi

⇓

Hello.java

```
public class Hello {
  public static void main(String[] args) {
    System.out.println("Hello.");           /* Output the message */
  }
}
```

Compile

`javac` : **Compile a source file of Java**

[How to compile] `javac␣<source␣file␣of␣Java>` ↵

ソースファイル

`Hello.java`

⇓

Compile

```
$ javac␣Hello.java ↵
$
```

⇓

Class file

`Hello.class`

Run/Validation

java : Run a class file

[How to run] `java␣<Class file (Without .class)>` ↵

Class file

Hello.class

⇓

Run

```
$ java␣Hello ↵
$
```

⇓

Validation/Completion

- If the result is incorrect, return to edit.

Template of a source file

Class name.java

```
import Library name;                        ⟵Optional

public class Class name {
  public static void main(String[] args) {

 Write a program here

  }
}
```

Note

- Uppercase and lowercase letters are distinguished.
- A class name starts with an uppercase letter. In this example, the class name is `Hello`.
- A file name is "`class name.java`", the source file name must match the class name in the program.

Notes to write a program (1)

Each statement ends with semicolons “;”.

```
System.out.println("Hello.");
```

```
int a = 1, b = 2, c;
c = a + b * 3;
c = (a + b) * 3;
```

Notes to write a program (2)

```
/*  Comment  */
//  Comment
```

:Write a comment/note

```
System.out.println("Hello.");    /* Output the message */
```

```
/*
    This is a comment.
    This is also a comment.
*/
```

```
// This is a comment.
   This is not a comment.
```

Notes to write a program (3)

To use line break and indentation make the program easy to read.

```
public class Hello {
  public static void main(String[] args) {
    System.out.println("Hello.");           /* Output the message */
  }
}
```

```
public class Hello { public static void main(String[] args) {
System.out.println("Hello."); /* Output the message */
}}
```

- The above programs are same. Both are correct.
- We can understand that it is easy to read an upper program.

Notes to write a program (4)

Common mistakes

- File name and Class name are different.
- There is not a semicolon at the end of a sentence.
- There are miss-spells.
- Correspondence of brackets
- There are full-pitch characters in a sentence.