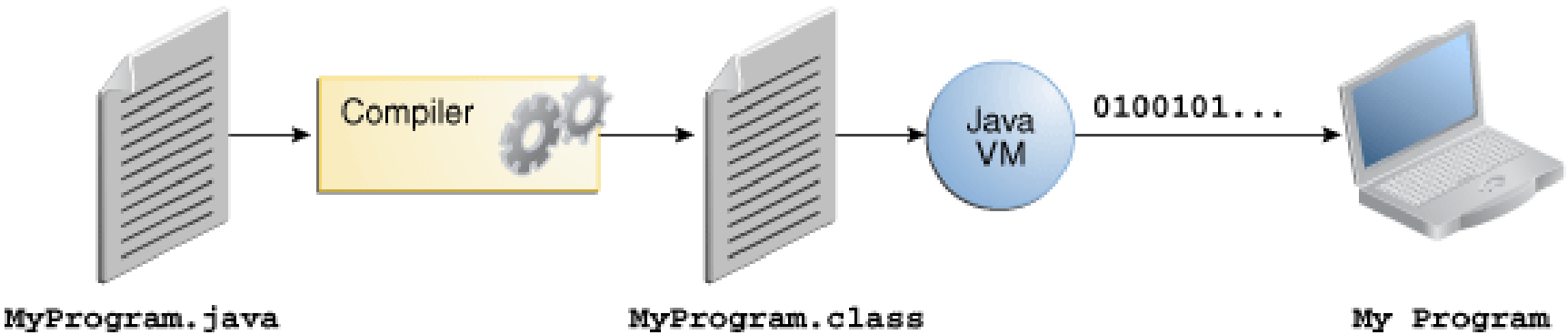
Programming Languages

Java Basics

Janyl Jumadinova

January 23 - 27, 2023

Java program development process



Simple first Java Program: "Hello World"

```
/** This is the first program people write in a new language,
 the "Hello World!".  In Java, this file must be named
 Welcome.java, with the first part of the name, Welcome, being
 the same as the name of the class. The filename itself
 (not the class name) must always end in .java to indicate
 to the operating system that it's a java source file.
*/
public class Welcome {
  public static void main ( String args[] ) {
      System.out.println ( "Hello World!" );
  }
 }
```

For today, try running this on
https://www.jdoodle.com/online-java-compiler/

Comments

Comments in Java can be one of three styles:

- **Single line:** starts at // anywhere on a line, ends at the end of that line
- **Multi-line:** starts with character sequence /* anywhere, ends with character sequence */ anywhere after that can span multiple lines
- **javadoc:** starts with character sequence /** anywhere, ends with character sequence */ anywhere, after that uses javadoc utility to create HTML documentation from code

- public class Welcome:
  - **public** means that something is available across packages (reserved word)
  - Name of the class has to be the same as the name of the .java file

- public class Welcome:
  - **public** means that something is available across packages (reserved word)
  - Name of the class has to be the same as the name of the .java file
- public static void main (String identifier[]):
  - The particular form of main is required by Java.
  - JVM starts executing here!
  - main is a static method, it is part of its class and not part of objects.
  - Strings in Java are sequence of characters

- public class Welcome:
  - **public** means that something is available across packages (reserved word)
  - Name of the class has to be the same as the name of the .java file
- public static void main (String identifier[]):
  - The particular form of main is required by Java.
  - JVM starts executing here!
  - main is a static method, it is part of its class and not part of objects.
  - Strings in Java are sequence of characters
- Braces { } are used to collect statements into a "block"

- public class Welcome:
  - **public** means that something is available across packages (reserved word)
  - Name of the class has to be the same as the name of the .java file
- public static void main (String identifier[]):
  - The particular form of main is required by Java.
  - JVM starts executing here!
  - main is a static method, it is part of its class and not part of objects.
  - Strings in Java are sequence of characters
- Braces { } are used to collect statements into a "block"
- Statements in Java end with semicolons.

Printing

- println: New line after printing
- print: No new line
- printf: Can specify format

Character Strings

string literal in class String

"ABC"

"This is interesting"

" "

"91"

Character Strings

string literal in class String

"ABC"
"This is interesting"
" "
"91"

- Use print or println methods to print a character string to the terminal
- System.out.println("CMPSC 111");
- the string "CMPSC 111" is a parameter: data sent to a method

String Concatenation

Appending one string to the end of another: use + operator

"This is "+ "interesting"

"Your grade is "+ "91"

String Concatenation

Appending one string to the end of another: use + operator

"This is "+ "interesting"
"Your grade is "+ "91"

- + is also used for arithmetic addition
- System.out.println("Adding "+ 12 + 23); is not the same as System.out.println("Adding "+ (12 + 23));

Escape Sequences

- Escape sequences, or escape characters, begin with a slash and are immediately followed by another character.
- This two-character sequence, inside " " allows you to control your output (\n, \t, \b) or output characters you wouldn't otherwise be able to (\\, \") inside a string.

Escape Sequences

Seq	Meaning	Example Code
\n	New line	System.out.println("Hi\nThere");
\t	Horizontal tab	System.out.println("What's\tup?");
\b	Backspace	System.out.println("Hi\b Hey");
\\	Backslash	System.out.println("Back\\Slash");
\"	Double quote	System.out.println("Dbl\"Quote");

Variables

- Variable is a name for a memory's location where a data value is stored.

Variables

- Variable is a name for a memory's location where a data value is stored.
- Variable Declaration allows the compiler to reserve space in the main memory that is large enough for the specified type

```
int count;
```

Variables

- Variable is a name for a memory's location where a data value is stored.
- Variable Declaration allows the compiler to reserve space in the main memory that is large enough for the specified type
  `int count;`
- Variable Assignment assigns a value to the variable
  `count = 0;`

Variables

- Variable is a name for a memory's location where a data value is stored.
- Variable Declaration allows the compiler to reserve space in the main memory that is large enough for the specified type
  `int count;`
- Variable Assignment assigns a value to the variable
  `count = 0;`
- Must give a value to the variable before using it in the `main` method.

Java Identifiers

- reserved keywords (class, public, static, void)
- Java classes, methods, variables: words we chose or make up when writing a program
  System, println, main, args

Java Identifiers

- reserved keywords (class, public, static, void)
- Java classes, methods, variables: words we chose or make up when writing a program
  System, println, main, args

Identifier

a letter followed by zero or more letters (including $ and _) and digits

Identifier Rules

- Identifiers must start with a letter, a currency character ($), or a connecting character such as the underscore (_).
- Identifiers cannot start with a number.
- After the first character, identifiers can contain any combination of letters, currency characters, connecting characters, or numbers.
- There is no limit to the number of characters an identifier can contain.
- You can't use a Java keyword as an identifier.
- Identifiers in Java are case-sensitive; foo and FOO are two different identifiers.

Data Types

- Data stored in memory is a string of bits (0 or 1)

Data Types

- Data stored in memory is a string of bits (0 or 1)
- How the computer interprets the string of bits depends on the context.
- In Java, we must make the context explicit by specifying the type of the data.

Data Types

- Java has two categories of data:
  primitive data (e.g., number, character)
  object data (programmer created types)
- There are 8 primitive data types: `byte`, `short`, `int`, `long`, `float`, `double`, `char`, `boolean`
- Primitive data are only single values; they have no special capabilities.

Primitive Data Types

- integers: byte, short, int, long
- floating point: float, double
- characters: char
- booleans: boolean

Common Primitive Data Types

Type	Description	Example of Literals
`int`	integers (whole numbers)	42, 60634, -8
`double`	real numbers	0.039, -10.2
`char`	single characters	'a', 'B', '&', '6'
`boolean`	logical values	true, false

Range of Values

Type	Storage	Range of Values
`int`	32 bits	-2,147,483,648 to 2,147,483,647
`double`	64 bits	$\pm 10^{-45}$ to $\pm 10^{38}$
`char`	16 bits = 2 bytes	0 to 2^{16} or \u0000 to \uFFFF
`boolean`	1 bit	NA