Introduction to Java Programming

"Basic programming constructs"

Advanced Programming

Shakirullah Waseeb
shakir.waseeb@gmail.com

Nangarhar University

February 21, 2017

Agenda

1. Operators
   - Arithmetic Operators
   - Decision Making
2. Classes, Objects, Methods
   - Class Declaration and Definition
   - Instantiation and Execution
3. Control Statement
   - Sequential, Selectional, and Repetition structures
   - Break and Continue statements
4. Questions and Discussion

Agenda

Arithmetic Operators

- Operators used for arithmetic calculations

Java operation	Operator	Algebraic expression	Java expression
Addition	+	$f + 7$	f + 7
Subtraction	-	$p - c$	p - c
Multiplication	*	bm	b * m
Division	/	x / y or $\frac{x}{y}$ or $x \div y$	x / y
Remainder	%	$r \bmod s$	r % s

[1]

Figure: arithmetic operators

Arithmetic Operators Precedence

- Precedence of arithmetic operators

Operator(s)	Operation(s)	Order of evaluation (precedence)
*	Multiplication	Evaluated first. If there are several operators of this type, they're evaluated from *left to right*.
/	Division	
%	Remainder	
+	Addition	Evaluated next. If there are several operators of this type, they're evaluated from *left to right*.
-	Subtraction	
=	Assignment	Evaluated last.

[1]

Figure: precedence of arithmetic operators

Agenda

Equality and relational operators

- condition is an expression that can be true or false
- e.g conditional expression in *if* selection statement, which make decision on condition's value
- Conditions in *if* statements can be formed using *equality* ($==, !=$) or *relational* ($>, <, >=, <=$) operators

Standard algebraic equality or relational operator	Java equality or relational operator	Sample Java condition	Meaning of Java condition
Equality operators			
=	==	x == y	x is equal to y
≠	!=	x != y	x is not equal to y
Relational operators			
>	>	x > y	x is greater than y
<	<	x < y	x is less than y
≥	>=	x >= y	x is greater than or equal to y
≤	<=	x <= y	x is less than or equal to y

[1]

Figure: precedence of arithmetic operators

Agenda

Class Declaration and Definition

- classes declared with *public* keyword must be:
  saved in a separate file
  file name must be same with class name

A simple class declaration example

```
public class SimpleClass {
                public void dispMessage(String str){
                        System.out.println(str);
                        }
                }
```

Save it as SimpleClass.java and compile it as javac SimpleClass.java

Agenda

Instantiation and Execution

- Every java application has a class that contain a *main* method, where application starts its execution
- Some programmers refer to such class as a driver class
- Example program containing *main* method

Example program

```
public class SimpleClassApp {
        public static void main (String args[]){
            SimpleClass sc = new SimpleClass();
            sc.dispMessage("A message from application");
           }
    }
```

Save it as SimpleClassApp.java, compile it using command javac SimpleClassApp.java, and finally run it as java SimpleClassApp.

Agenda

Sequential, Selectional, and Repetition structures

- Control structure:
  sequential execution; execute in the order in which program is written
  transfer of control; specify which instruction to execute next
- Sequence Structure
  - normal execution of program instruction in the order they are written
- Selection Structure
  - single selection statements (*if* statement)
  - double selection statements (*if .. else* statement)
  - multiple selection statements (*switch* statement)
- Repetition Structure
  - also called looping statements
  - looping continuation condition
  - *while*, *do while*, and *for*

Agenda

Break and Continue statements

- *break* statement:
  when executed in a *while, for, do...while or switch*, causes immediate exit from that statement
  typically use to escape early from a loop or to skip the remainder of a *switch*
- *continue* statement:
  when executed in a *while, for or do...while*, skips the remaining statements in the loop body and proceeds with the *next iteration* of the loop
  while and do...while: immediately test loop-continuation
  for: increment expression executes, then loop-continuation is tested

Your Turn: Time to hear from you!

1

[1] https://fensafitters.files.wordpress.com/2013/07/3d095.jpg

References


P.J. Deitel, H.M. Deitel
Java How to program, 10th Edition .
Prentice Hall, 2015.


P.J. Deitel, H.M. Deitel
Java How to program, 9th Edition .
Prentice Hall, 2012.


Herbert Schildt
The complete reference Java2, 5th Edition .
McGraw-Hill/Osborne, 2002.