

CSC 125

Object Oriented Programming

Ch06_Methods

Dr. Fadi Alzhouri



What is a Method?

- A set of statements to perform a specific task.
- A block of code which only runs when it is called.
- aka: **functions**, modules, **procedures**, subroutine, or subprograms)
- **Why?**: To encapsulate functionality and promote code reuse.
 - It is a technique of writing a piece of code once and using it multiple times.
 - Avoiding repetition
- Programs can be quite large; we need to break them down into smaller functions (methods).
- Methods call other methods to complete specific tasks.

A view of methods

```
public class Main
{
    public static int add(int a, int b) {
        // Method 1 body
    }

    public static int subtract(int a, int b) {
        // Method 2 body
    }

    public static void main(String[] args) {
        Statement 1;
        int sum1 = add(num1, num2); // Calling method 1
        Statement 3;
        int diff = subtract(num1, num2); // Calling method 2
        Statement 5;
        int sum2 = add(num2, num3); // Calling method 1
    }
}
```



Using Methods

- Method's motivations?
- **Divide and Conquer**: Divide the problem into smaller pieces, and you conquer the complexity of the problem.
- **Reusability**: Can be used in more than one place in a program or in different programs.
- **Simplicity**: Simplify code maintenance.

Methods Types

1) **Predefined** Methods (Built-in)

- Methods that the Java **API** provides.
- Examples:
 - `System.out.println()`
 - `String.length()`

2) **User-Defined** Methods

- Void Methods (nonvalue-returning): Methods that **do not return** a value.
- Non-Void Methods (Value-returning): Methods that **return** a value.
 - 1) have a data type
 - 2) return only one value (thing) to the caller

Predefined Methods (Built-in)

- They simplify programming by offering commonly used functionalities.
- Example I: **Math methods**
 - The Math class provides methods for performing basic numeric operations.

Method	Description
<code>exp(double a)</code>	Returns Euler's number e raised to the power of the specified double value.
<code>pow(double a, double b)</code>	Returns the value of the first argument raised to the power of the second argument.
<code>sqrt(double a)</code>	Returns the square root of the specified double value. If the value is negative, NaN is returned.
<code>ceil(double a)</code>	Returns the smallest (closest to negative infinity) double value that is greater than or equal to the argument and is equal to a mathematical integer.

Predefined Methods (Built-in) (cont.)

Method	Description
<code>floor(double a)</code>	Returns the largest (closest to positive infinity) double value that is less than or equal to the argument and is equal to a mathematical integer.
<code>round(double a)</code>	Returns the closest integer to the argument, rounding up if the fractional part is 0.5 or greater.
<code>random()</code>	Returns a double value greater than or equal to 0.0 and less than 1.0, representing a pseudo-random number.

Exercise:

```
public class Main
{
    public static void main(String[] args) {

        System.out.println(Math.pow(2.0, 3.0));

        double sqrtResult = Math.sqrt(16.0);
        System.out.println("Math.sqrt(16.0) = " + sqrtResult);

        long roundResult = Math.round(3.5);
        System.out.println("Math.round(3.5) = " + roundResult);

        System.out.println(Math.random());
    }
}
```

```
8.0
Math.sqrt(16.0) = 4.0
Math.round(3.5) = 4
0.8661053575567472
```


Predefined Methods (Built-in) (cont.)

- Example I: **String methods**
 - The String class provides methods for manipulating strings.
 - We discussed several String methods in Chapter 5.
 - Such as:
 - `length()`
 - `charAt()`

```
String text = "Welcome to Java";  
System.out.println(text.length());  
System.out.println(text.charAt(3));
```



```
15  
c
```

User-Defined Methods

-

References

- **Introduction to Java Programming, Brief Version, Global Edition, 11th edition**, Published by Pearson (June 21, 2018) © 2018