6

Methods: A Deeper Look

Exam 1

- Wednesday, April 8
- in-class (5% of overall grade)
- practice exams will be available today
- review of practice exams on Monday, April 6
- covers Chap. 1 – 6
- some theory, emphasis on coding practice

Last time

- switch, break, continue statements
- finished overview of control statements
- GUI: colors and Unicode
- gradient image example

Objectives

- Chapter 6: all about methods
- static methods and static fields
- constants (use `final` keyword)
- mathematical methods (`java.lang.Math`)
- random number generation (`java.util.Random`)
- practice code: using Math and Random packages

Outline

Outline

Chapter 6 overview

- **Methods – units of code to perform some function**
- **`static` methods can be called without the need for an object of the class**
- **`final` keyword to declare something constant**
- **Random number generation**

Methods in Java

- **Methods**
  - **Called functions or procedures in some other languages**
  - **Modularize programs by separating its tasks into self-contained units**
  - **Enable a divide-and-conquer approach**
  - **Are reusable in later programs**
  - **Prevent repeating code**

`static` methods

- **`static` method (or class method)**
  - **Applies to the class as a whole instead of a specific object of the class**
  - **Call a `static` method by using the method call:**
    ClassName **.** ***methodName*** **(** ***arguments*** **)**
  - **All methods of the `Math` class are `static`**
  - **Examples:**
    - `Math.sqrt( 900.0 )`
    - `Math.cos( 1.57 )`

Software Engineering Observation 6.4

Class `Math` is part of the `java.lang` package, which is implicitly imported by the compiler, so it is not necessary to import class `Math` to use its methods.

Method	Description	Example
`abs( x )`	absolute value of *x*	`abs( 23.7 )` is 23.7 `abs( 0.0 )` is 0.0 `abs( -23.7 )` is 23.7
`ceil( x )`	rounds *x* to the smallest integer not less than *x*	`ceil( 9.2 )` is 10.0 `ceil( -9.8 )` is -9.0
`cos( x )`	trigonometric cosine of *x* (*x* in radians)	`cos( 0.0 )` is 1.0
`exp( x )`	exponential method *ex*	`exp( 1.0 )` is 2.71828 `exp( 2.0 )` is 7.38906
`floor( x )`	rounds *x* to the largest integer not greater than *x*	`Floor( 9.2 )` is 9.0 `floor( -9.8 )` is -10.0
`log( x )`	natural logarithm of *x* (base *e*)	`log( Math.E )` is 1.0 `log( Math.E * Math.E )` is 2.0
`max( x, y )`	larger value of *x* and *y*	`max( 2.3, 12.7 )` is 12.7 `max( -2.3, -12.7 )` is -2.3
`min( x, y )`	smaller value of *x* and *y*	`min( 2.3, 12.7 )` is 2.3 `min( -2.3, -12.7 )` is -12.7
`pow( x, y )`	*x* raised to the power *y* (i.e., *xy*)	`pow( 2.0, 7.0 )` is 128.0 `pow( 9.0, 0.5 )` is 3.0
`sin( x )`	trigonometric sine of *x* (*x* in radians)	`sin( 0.0 )` is 0.0
`sqrt( x )`	square root of *x*	`sqrt( 900.0 )` is 30.0
`tan( x )`	trigonometric tangent of *x* (*x* in radians)	`tan( 0.0 )` is 0.0

Fig. 6.2 | Math class methods.

`static` method example: main( )

- **Method `main`**
  - **`main` is declared `static` so it can be invoked without creating an object of the class containing `main`**
  - **Any class can contain a `main` method, but usually want to create "testing" classes that would invoke the main method**

`static` fields

- **`static` fields (or class variables)**
  - **Are fields where one copy of the variable is shared among all objects of the class**
- **Constants**
  - **Keyword `final`**
  - **Cannot be changed after initialization**
- **`Math.PI` and `Math.E` are `final static` fields of the `Math` class**

Notes on Declaring and Using Methods

- **Three ways to call a method:**
  - **Use a method name by itself to call another method of the same class**
  - **Use a variable containing a reference to an object, followed by a dot (.) and the method name to call a method of the referenced object**
  - **Use the class name and a dot (.) to call a `static` method of a class**
- **`static` methods cannot call non-`static` methods and variables of the same class directly**

Random number generation

- **Random number generation**
  - **`static` method `random` from class `Math`**
    - **Returns `doubles` in the range `0.0` <= x < `1.0`**
  - **class `Random` from package `java.util`**
    - **Can produce pseudorandom `boolean`, `byte`, `float`, `double`, `int`, `long` and Gaussian values**
    - **Is seeded with the current time of day to generate different sequences of numbers each time the program executes**