

CISC 3115 EWQ6

Java API Classes: Date, Random, Math, and Point2D

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Outline

- Just discussed
 - Default constructor
 - Accessing objects via reference variables
 - Primitive and reference variables
 - Garbage collection
- A few classes in the Java Library (Java API)
 - Date, Random, Point2D, Math

Java API and Library

- Java API: Java Application Programming Interface
 - Define the interface with which an application interacts with Java
 - Classes and methods that an application programmer can use in their own programs
- Java Library: implementation of the classes and methods

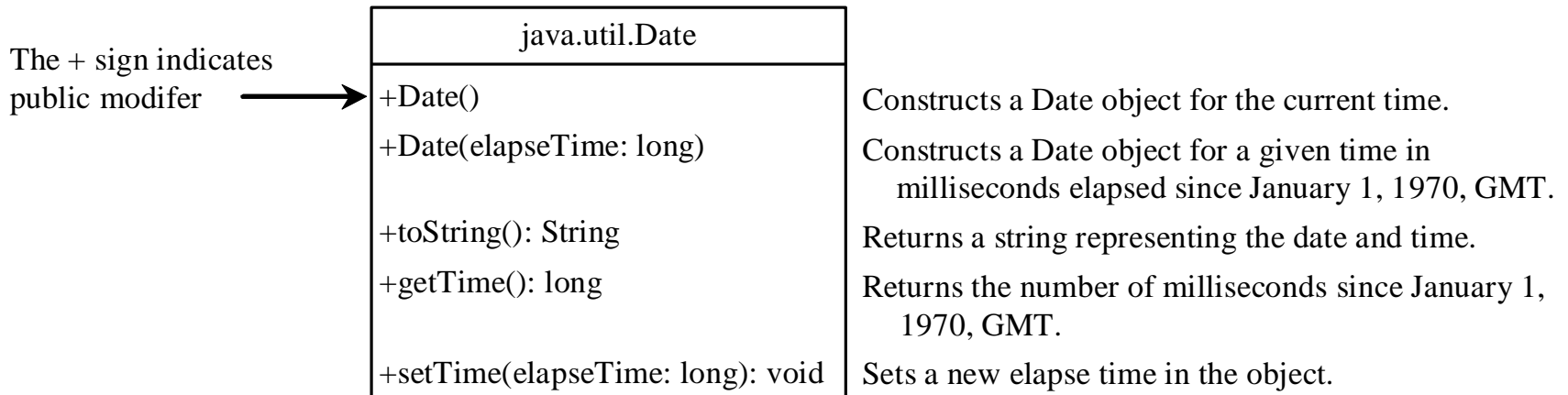
A Few Classes in Java Library

- Date
- Random
- Point2D

The Date Class

- A system-independent encapsulation of date and time in the java.util.Date class.
- Represent a specific instant in time, with millisecond precision
- Example usage:
 - You can use the Date class to create an instance for the current date and time and use its toString method to return the date and time as a string.
- API documentation (may be intimidating to some, a good read nonetheless)
 - <https://docs.oracle.com/javase/8/docs/api/java/util/Date.html>

The Date Class: UML Class Diagram



- Why “public”?

The Date Class: Example

- Showing current system date and time
 - the import statement
 - Java package: java.util is a package while Date is a class in the package

```
PrintDate.java x
1 import java.util.Date;
2
3 class PrintDate {
4     public static void main(String[] args) {
5         Date date = new Date();
6         System.out.println("The date is " + date.toString());
7     }
8 }
```

The Random Class

- A pseudo-random number generator in the `java.util.Random` class
- Use it to generate a sequence of pseudo-random numbers
- Example usage:
 - Generate a sequence random integers
 - Generate a sequence random float point numbers
 - Generate a sequence random Boolean values
- API documentation (may be intimidating to some, a good read nonetheless)
 - <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>

The Random Class: UML Class Diagram

java.util.Random	
+Random()	Constructs a Random object with the current time as its seed.
+Random(seed: long)	Constructs a Random object with a specified seed.
+nextInt(): int	Returns a random int value.
+nextInt(n: int): int	Returns a random int value between 0 and n (exclusive).
+nextLong(): long	Returns a random long value.
+nextDouble(): double	Returns a random double value between 0.0 and 1.0 (exclusive).
+nextFloat(): float	Returns a random float value between 0.0F and 1.0F (exclusive).
+nextBoolean(): boolean	Returns a random boolean value.

The Random Class: Example

- What is a “seed”?

```
RandomNumbers.java x
1 import java.util.Random;
2
3 class RandomNumbers {
4     public static void main(String[] args) {
5         Random random1 = new Random(3);
6         System.out.print("From random1: ");
7         for (int i = 0; i < 10; i++) {
8             System.out.print(random1.nextInt(1000) + " ");
9         }
10
11        Random random2 = new Random(3);
12        System.out.print("\nFrom random2: ");
13        for (int i = 0; i < 10; i++) {
14            System.out.print(random2.nextInt(1000) + " ");
15        }
16
```

```
17        Random random3 = new Random(4);
18        System.out.print("\nFrom random3: ");
19        for (int i = 0; i < 10; i++) {
20            System.out.print(random3.nextInt(1000) + " ");
21        }
22
23        Random random4 = new Random();
24        System.out.print("\nFrom random4: ");
25        for (int i = 0; i < 10; i++) {
26            System.out.print(random4.nextInt(1000) + " ");
27        }
28
29        Random random5 = new Random();
30        System.out.print("\nFrom random5: ");
31        for (int i = 0; i < 10; i++) {
32            System.out.print(random5.nextInt(1000) + " ");
33        }
34    }
35 }
```

Pseudo-Random Numbers

- The Random class generates pseudo-random numbers, i.e., the numbers are generated by an algorithm
- Implication
 - They are in fact deterministic although appear random.
 - Given two identical seeds, the sequences of “random” numbers are identical as well

Math.random()

- The Math class in the Java Library has a random method
 - Generating pseudo-random double values in interval [0.0, 1.0)
 - Described in the API documentation
 - [https://docs.oracle.com/javase/10/docs/api/java/lang/Math.html#random\(\)](https://docs.oracle.com/javase/10/docs/api/java/lang/Math.html#random())

“When this method is first called, it creates a single new pseudorandom-number generator, exactly as if by the expression

```
new java.util.Random()
```

This new pseudorandom-number generator is used thereafter for all calls to this method and is used nowhere else. “

The Point2D Class

- Representing a point in a two-dimensional plane in the `javafx.geometry.Point2D` class.
 - A point is represented by its coordinates (x, y) in the plane
- Example usage:
 - You can use it to compute distance between two points, and more
- API documentation (may be intimidating to some, a good read nonetheless)
 - <https://docs.oracle.com/javase/8/javafx/api/javafx/geometry/Point2D.html>

The Point2D Class: UML Class Diagram

javafx.geometry.Point2D

```
+Point2D(x: double, y: double)
+distance(x: double, y: double): double
+distance(p: Point2D): double
+getX(): double
+getY(): double
+toString(): String
```

Constructs a `Point2D` object with the specified *x*- and *y*-coordinates.

Returns the distance between this point and the specified point (*x*, *y*).

Returns the distance between this point and the specified point *p*.

Returns the *x*-coordinate from this point.

Returns the *y*-coordinate from this point.

Returns a string representation for the point.

The Point2D Class: Example

```
TestPoint2D.java
1 import java.util.Scanner;
2 import javafx.geometry.Point2D;
3
4 class TestPoint2D {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner (System.in);
7
8         System.out.println("Enter point1's x=, and y-coordinates: ");
9         double x1 = sc.nextDouble();
10        double y1 = sc.nextDouble();
11
12        System.out.println("Enter point2's x=, and y-coordinates: ");
13        double x2 = sc.nextDouble();
14        double y2 = sc.nextDouble();
15
16        Point2D p1 = new Point2D(x1, y1);
17        Point2D p2 = new Point2D(x2, y2);
18        System.out.println("p1 is " + p1.toString());
19        System.out.println("p2 is " + p2.toString());
20        System.out.println("The distance between p1 and p2 is " + p1.distance(p2));
21        System.out.println("The midpoint between p1 and p2 is " + p1.midpoint(p2).toString());
22    }
23 }
```

Test JavaFX applications

```
MINGW64:/c/Users/hui/CISC3115/C0907
hui@N-1432-59788 MINGW64 ~/CISC3115/C0907
$ javac -version
javac 17.0.2

hui@N-1432-59788 MINGW64 ~/CISC3115/C0907
$ java -version
openjdk version "17.0.2" 2022-01-18
OpenJDK Runtime Environment (build 17.0.2+8-86)
OpenJDK 64-Bit server VM (build 17.0.2+8-86, mixed mode, sharing)

hui@N-1432-59788 MINGW64 ~/CISC3115/C0907
$ ls /c/Applications/java/javafx-sdk-18.0.2/
bin/  legal/  lib/  src.zip

hui@N-1432-59788 MINGW64 ~/CISC3115/C0907
$ javac --module-path "/c/Applications/java/javafx-sdk-18.0.2/lib/" --add-modules
s javafx.graphics Point2DClient.java

hui@N-1432-59788 MINGW64 ~/CISC3115/C0907
$ java --module-path "/c/Applications/java/javafx-sdk-18.0.2/lib/" --add-modules
javafx.graphics Point2DClient.java
pt1 is at Point2D [x = 1.0, y = 1.0]
pt2 is at Point2D [x = 0.0, y = 0.0]
The distance between pt1 and pt2 is 1.4142135623730951

hui@N-1432-59788 MINGW64 ~/CISC3115/C0907
$ |
```


Questions?

- Concept of Java API and Java Library
- A few classes in the Java Library
 - Date
 - Random
 - Point2D
 - Math
- In which Java packages are they?