CSCI1530 Computer Principles and Java Programming

Tutorial 4

Scanner & Math methods

Zheng Qingqing
SHB 911
qqzheng@cse.cuhk.edu.hk

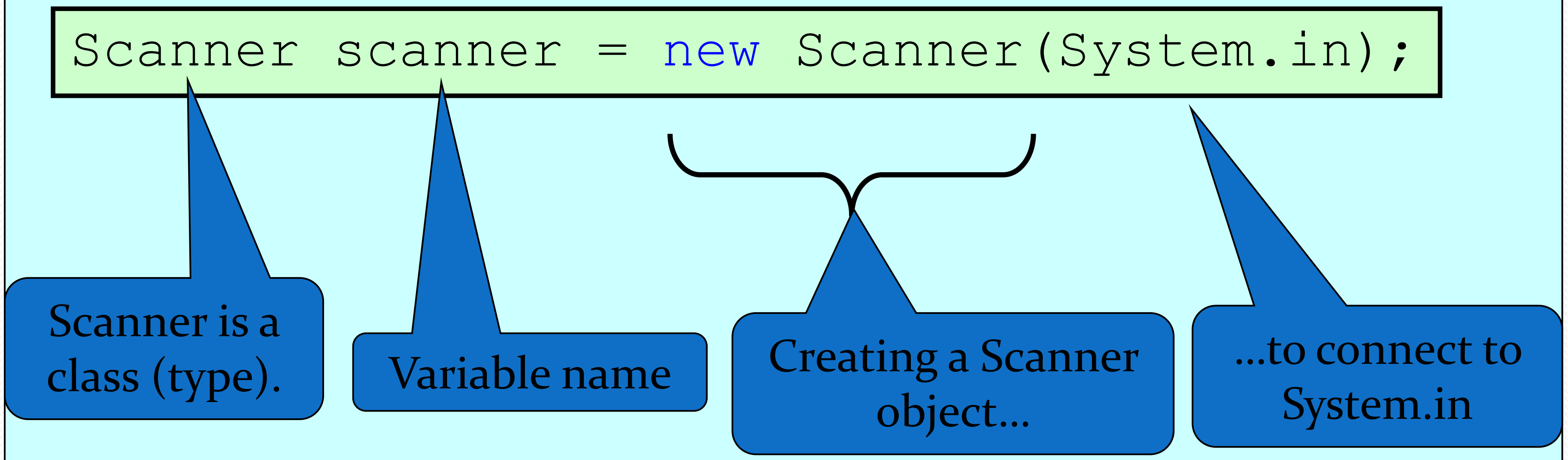
Content

- Using Scanner class
- Java API Specification
- Using Math class in calculations
- JOptionPane (Assignment 2)

Using Scanner class

- Using Scanner with System.in, we could input values from keyboard.

Basic use of `Scanner` with `System.in`



```
Scanner scanner = new Scanner(System.in);
```

Set up a scanner to read values from keyboard.

Details in Lecture 2

Example from Lecture 2

(Line 9) Execution is *paused* when **`scanner.nextInt()`** is called, expecting an integer input from the user.

```
1   import java.util.*;
2
3   class Example {
4      public static void main(String[] args) {
5         int num1;
6         Scanner scanner = new Scanner(System.in);
7
8         System.out.println("Please enter an integer:");
9         num1 = scanner.nextInt();
10        System.out.println("num1 = " + num1);
11     }
12  }
```

Must include this line in order to use Scanner.

```
Please enter an integer:
123↵
num1 = 123
```

Read inputs

- Read in a **double** value

```
double num1 = scanner.nextDouble();
```

- Read in an **integer** value

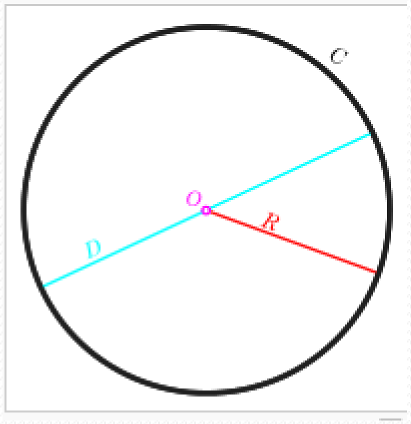
```
int num2 = scanner.nextInt();
```

Same scanner object can be used in different read statements

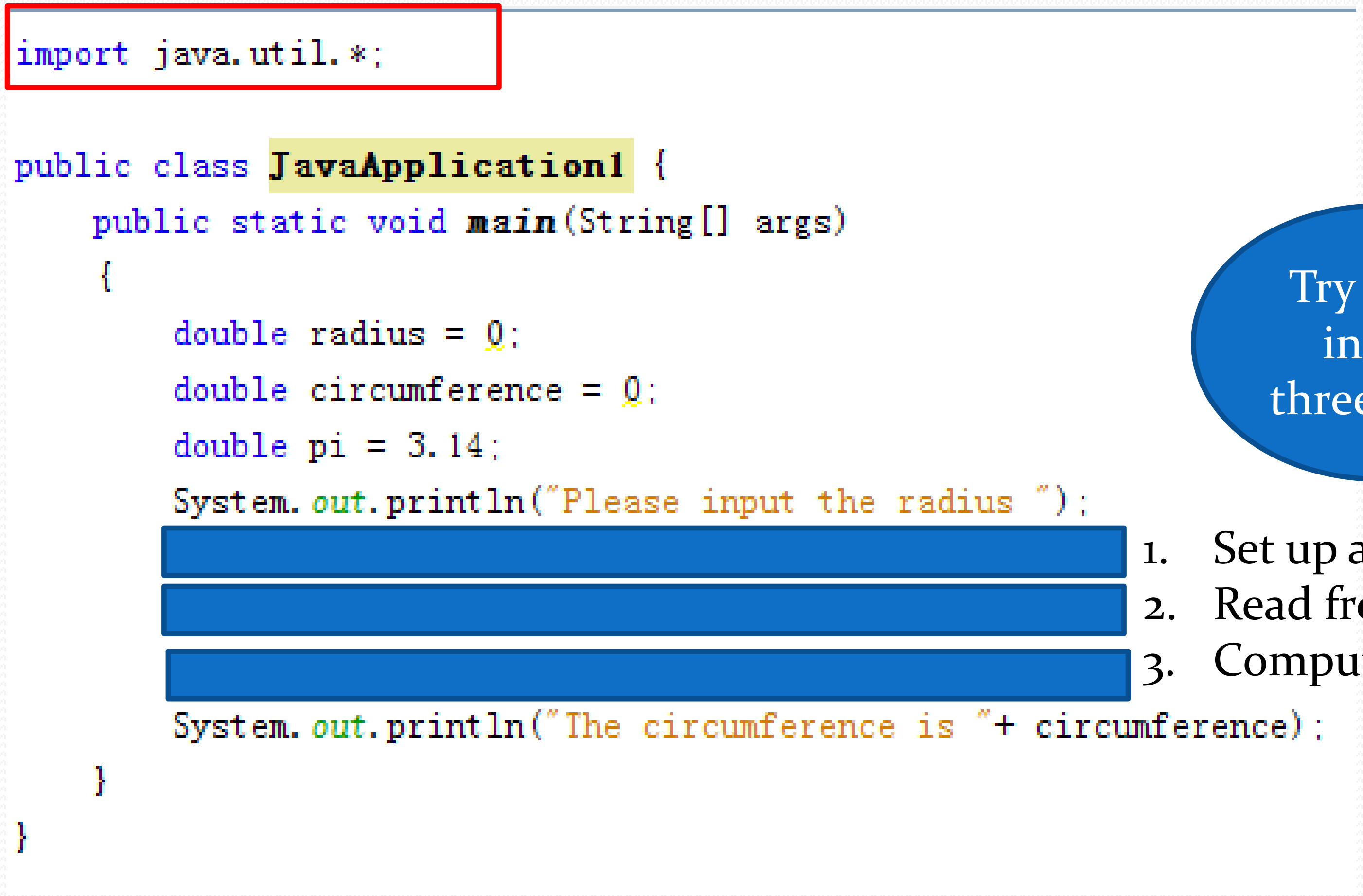
Practice

To write a simple program aiming to:

Compute the circumference of a circle, when given the radius from the keyboard



Given radius, output circumference

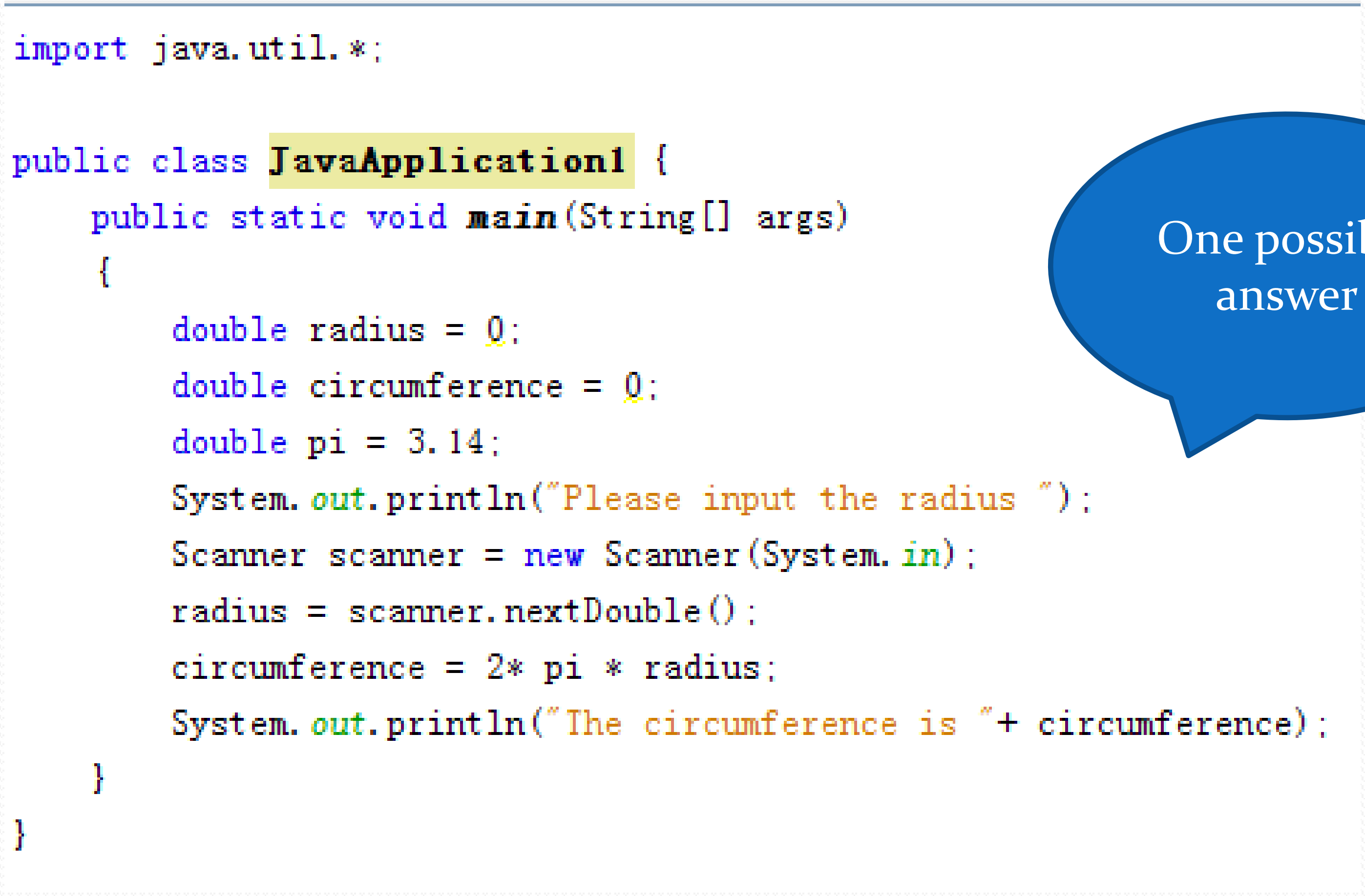


```
import java.util.*;

public class JavaApplication1 {
    public static void main(String[] args)
    {
        double radius = 0;
        double circumference = 0;
        double pi = 3.14;
        System.out.println("Please input the radius ");



        System.out.println("The circumference is "+ circumference);
    }
}
```

Try to fill in the three lines.

1. Set up a scanner
2. Read from keyboard
3. Compute circumference

- Given radius, output circumference



```
import java.util.*;

public class JavaApplication1 {
    public static void main(String[] args)
    {
        double radius = 0;
        double circumference = 0;
        double pi = 3.14;
        System.out.println("Please input the radius ");
        Scanner scanner = new Scanner(System.in);
        radius = scanner.nextDouble();
        circumference = 2* pi * radius;
        System.out.println("The circumference is "+ circumference);
    }
}
```

One possible answer

Java API document

Using Predefined Classes & Methods

- There are a lot of predefined classes and methods provided for you in Java.
- To call a pre-defined method, you need to know the following information about the method
  - Name
  - Functionality
  - Parameters
  - Return value

int nextInt()

Name: nextInt
Functionality: to get an integer from the keyboard
Parameter: no parameter
Return value: an integer from input

Using Predefined Classes & Methods

- You may also need to know which *package(s)* to import
  - e.g.: To use methods in Scanner class, you should include

    ```
    import java.util.*; (or)
    import java.util.Scanner;
    ```

    at the beginning of your java program

Using Predefined Classes & Methods

- Nobody wants to store all those methods in your brain.

- How to know these information?
    - Turn to the **Java API Specification**:

    A document which detailed illustrates all pre-defined classes and methods provided by Java

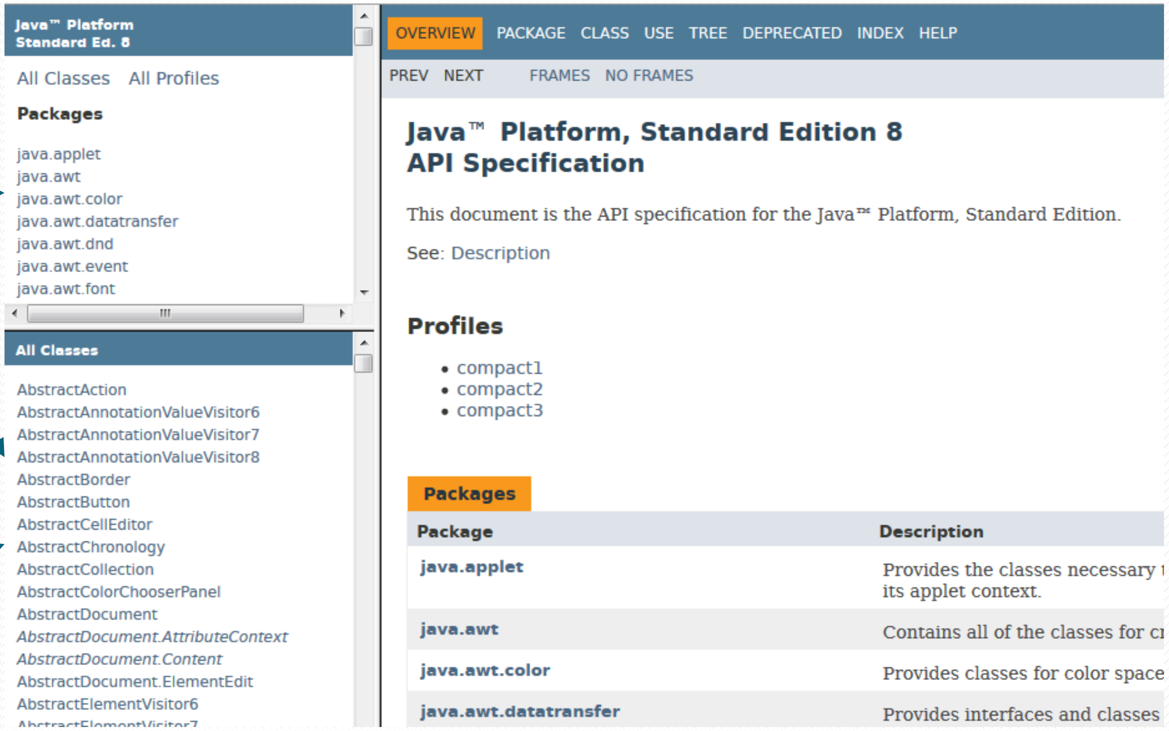
    Really helpful: In general cases, you don't need to type in questions like "How to input integer in Java" in search engine, but just search the API Specification.

Java API Specification

- http://docs.oracle.com/javase/7/docs/api/ or
- http://docs.oracle.com/javase/8/docs/api/

Choose a package, then choose the class in that package

OR lookup the class from all classes


Java™ Platform
Standard Ed. 8

All Classes All Profiles

Packages

java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font

All Classes

AbstractAction
AbstractAnnotationValueVisitor6
AbstractAnnotationValueVisitor7
AbstractAnnotationValueVisitor8
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractChronology
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeContext
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractElementVisitor6

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

Java™ Platform, Standard Edition 8
API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

See: Description

Profiles

- compact1
- compact2
- compact3

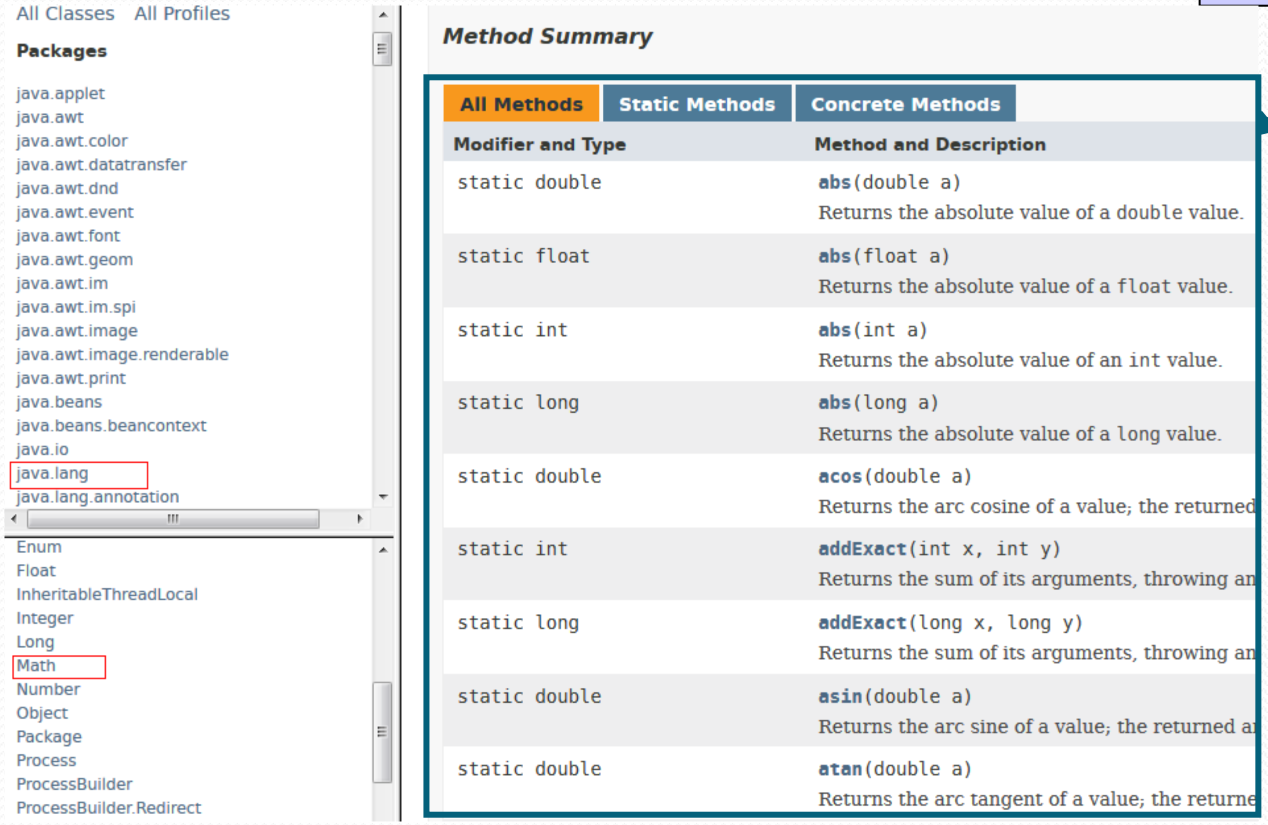
Packages

Package	Description
java.applet	Provides the classes necessary its applet context.
java.awt	Contains all of the classes for c
java.awt.color	Provides classes for color space
java.awt.datatransfer	Provides interfaces and classes

Java API Specification

- For example , look up methods in Math Class, a class providing math methods.

Summary of all the methods in Math Class



All Classes All Profiles

Packages

java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font
java.awt.geom
java.awt.im
java.awt.im.spi
java.awt.image
java.awt.image.renderable
java.awt.print
java.beans
java.beans.beancontext
java.io
java.lang
java.lang.annotation

Enum
Float
InheritableThreadLocal
Integer
Long
Math
Number
Object
Package
Process
ProcessBuilder
ProcessBuilder.Redirect

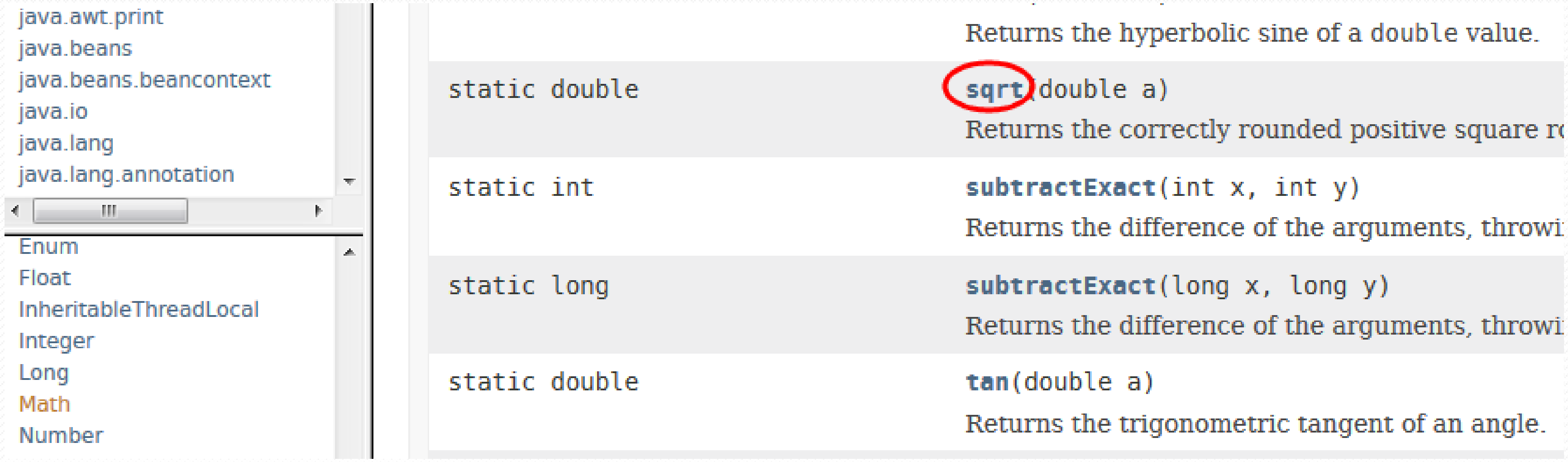
Method Summary

All Methods | Static Methods | Concrete Methods

Modifier and Type	Method and Description
`static double`	`abs(double a)` Returns the absolute value of a double value.
`static float`	`abs(float a)` Returns the absolute value of a float value.
`static int`	`abs(int a)` Returns the absolute value of an int value.
`static long`	`abs(long a)` Returns the absolute value of a long value.
`static double`	`acos(double a)` Returns the arc cosine of a value; the returned
`static int`	`addExact(int x, int y)` Returns the sum of its arguments, throwing an
`static long`	`addExact(long x, long y)` Returns the sum of its arguments, throwing an
`static double`	`asin(double a)` Returns the arc sine of a value; the returned a
`static double`	`atan(double a)` Returns the arc tangent of a value; the returne

Java API Specification

- You can see the details of a method by clicking the name of the corresponding method



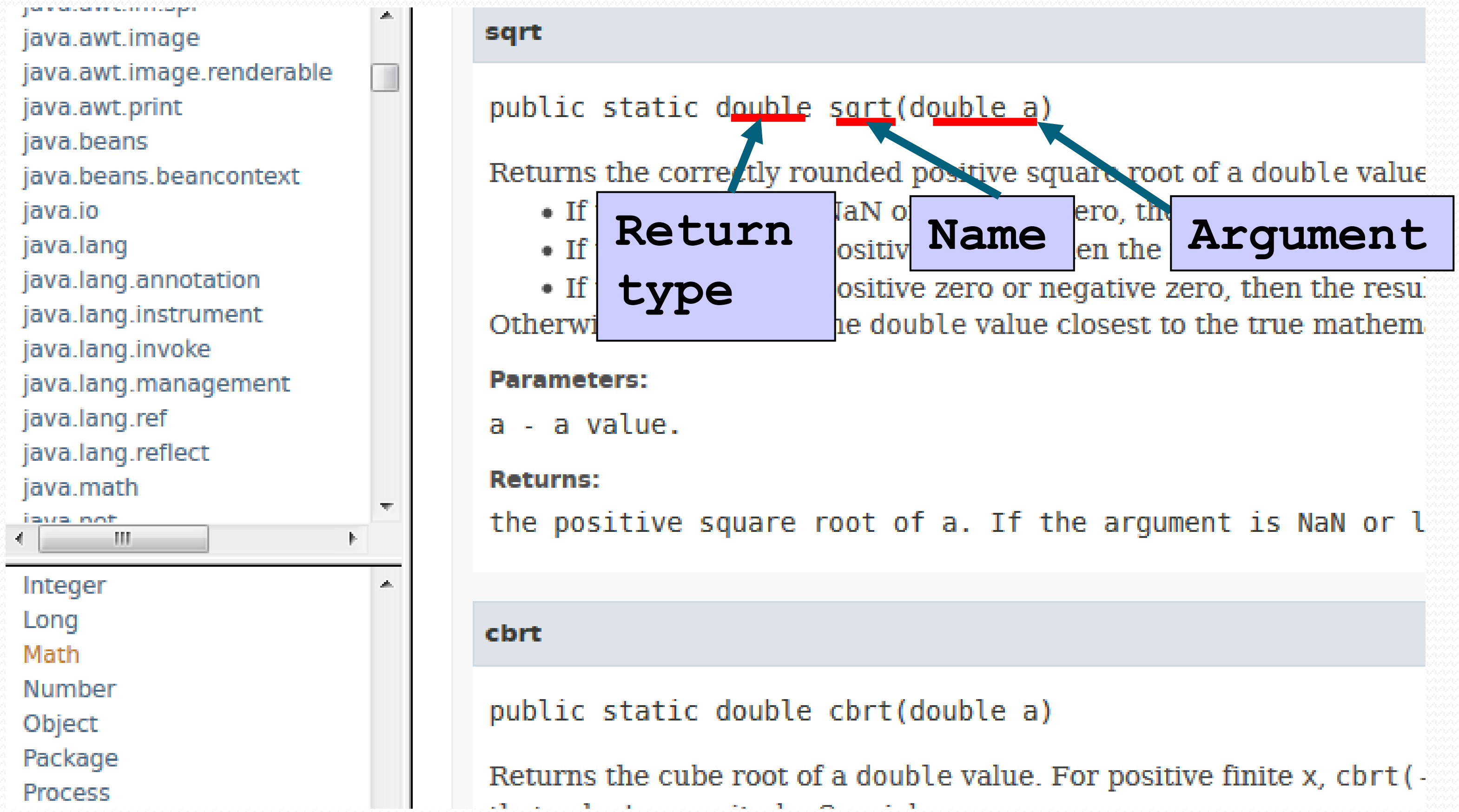
java.awt.print
java.beans
java.beans.beancontext
java.io
java.lang
java.lang.annotation

Enum
Float
InheritableThreadLocal
Integer
Long
Math
Number

	Returns the hyperbolic sine of a double value.
static double	sqrt(double a) Returns the correctly rounded positive square r
static int	subtractExact(int x, int y) Returns the difference of the arguments, throwi
static long	subtractExact(long x, long y) Returns the difference of the arguments, throwi
static double	tan(double a) Returns the trigonometric tangent of an angle.

Java API Specification

- Details of sqrt() method in Math Class



java.awt.image
java.awt.image.renderable
java.awt.print
java.beans
java.beans.beancontext
java.io
java.lang
java.lang.annotation
java.lang.instrument
java.lang.invoke
java.lang.management
java.lang.ref
java.lang.reflect
java.math

Integer
Long
Math
Number
Object
Package
Process

sqrt

```
public static double sqrt(double a)
```

Returns the correctly rounded positive square root of a double value

- If
- If
- If ositive zero or negative zero, then the resu

Otherwi he double value closest to the true mathem

Parameters:

`a - a value.`

Returns:

`the positive square root of a. If the argument is NaN or l`

cbrt

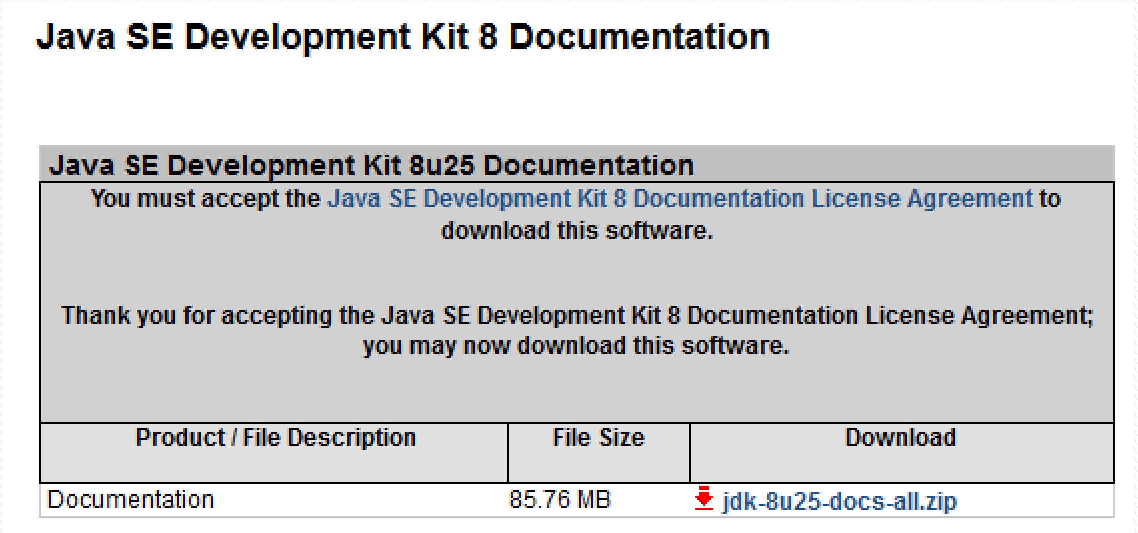
```
public static double cbrt(double a)
```

Returns the cube root of a double value. For positive finite x, cbrt(-

Download Java API Specification

If needed, you can download the document to your notepad from oracle so that you can look up things on it when you are away from Internet

http://www.oracle.com/technetwork/java/javase/documentation/jdk8-doc-downloads-2133158.html



Java SE Development Kit 8 Documentation

Java SE Development Kit 8u25 Documentation

You must accept the Java SE Development Kit 8 Documentation License Agreement to download this software.

Thank you for accepting the Java SE Development Kit 8 Documentation License Agreement; you may now download this software.

Product / File Description	File Size	Download
Documentation	85.76 MB	jdk-8u25-docs-all.zip

Using Math class in calculations

Using Math class in calculations

- Math class contains predefined methods for performing basic numeric operations
  - Square root
  - Logarithm
  - Exponential, etc
- You don't have to implement them again when you need to use them
  - They can be used by calling them directly

Example

```java
class MathExample {
  public static void main(String[] args) {
  double x;

  x = Math.sqrt(10);
  System.out.println("x = square root of 10 = " + x);
  System.out.println("The ceiling of x = "
                          + Math.ceil(x));
  System.out.println("2 to the power of x = "
                          + Math.pow(2, x));
}
}
```

Return value can be stored in a variable or used in an expression

```
x = square root of 10 = 3.1622776601683795
The ceiling of x = 4.0
2 to the power of x = 8.952419619470874
```

Some methods in Math Class

The return data type remains "double"

Mathods	Description	Examples
`ceil( x )`	rounds *x* to the smallest integer not less than *x* $\lceil X \rceil$	`Math.ceil(9.2)` is 10.0 `Math.ceil(-9.8)` is -9.0
`floor ( x )`	rounds *x* to the largest integer not greater than *x* $\lfloor X \rfloor$	`Math.floor(9.2)` is 9.0 `Math.floor(-9.8)` is -10.0
`exp( x )`	exponential function e^x	`Math.exp(1.0)` is 2.71828
`abs( x )`	absolute value of *x* $\|x\|$	`Math.abs(5.1)` is 5.1 `Math.abs(0.0)` is 0.0 `Math.abs(-8.76)` is 8.76

Some methods in Math Class

Mathods	Description	Examples

`pow( x, y )`	*x* raised to power *y*
	X^y
		`Math.pow(2, 7)`
		`is 128.0`
		`Math.pow(9, .5)`
		`is 3.0`

`sqrt( x )`	square root of *x*
	$\sqrt{x}$
		`Math.sqrt(900.0)`
		`is 30.0`
		`Math.sqrt(9.0)`
		`is 3.0`

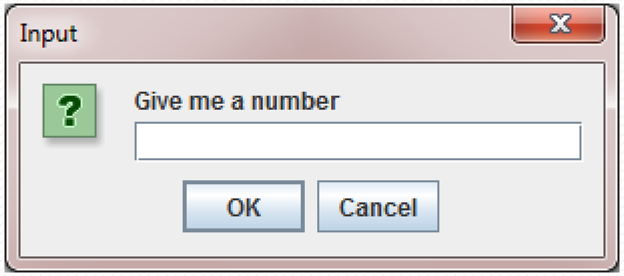
`log(x)`	natural logarithm of *x* (base *e*)
	$\log_e x$ or $\ln x$
	`ln e = 1`
	$\ln e^x = x * \ln e = x$
		`Math.log(2.718282)`
		`≈ 1.0`
		`Math.log(exp(3.0))`
		`is 3.0`

Tips for Assignment 2 JOptionPane

Input -- JOptionPane vs Console

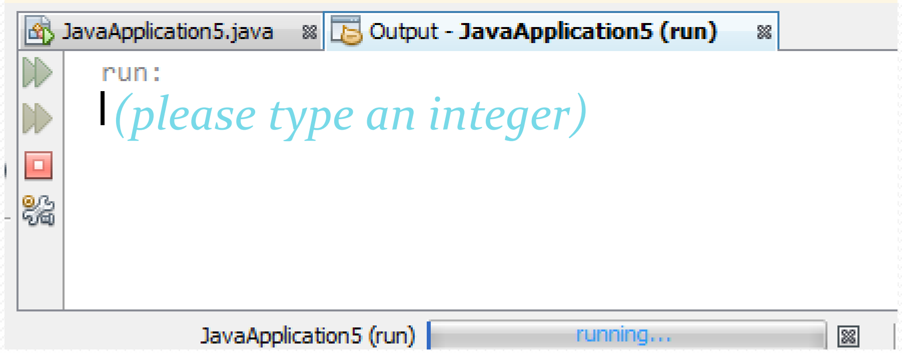
- JOptionPane

```
String text = JOptionPane.showInputDialog(null, "Give me a number");
```



- Console

```
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
```



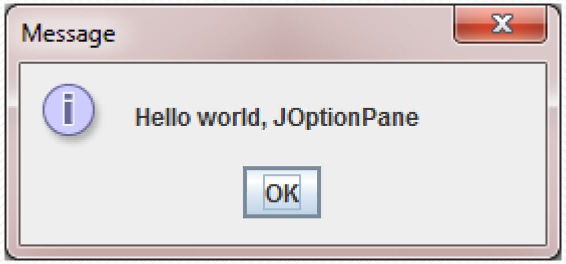
```
run:
```

(please type an integer)

Output -- JOptionPane vs Console

- JOptionPane

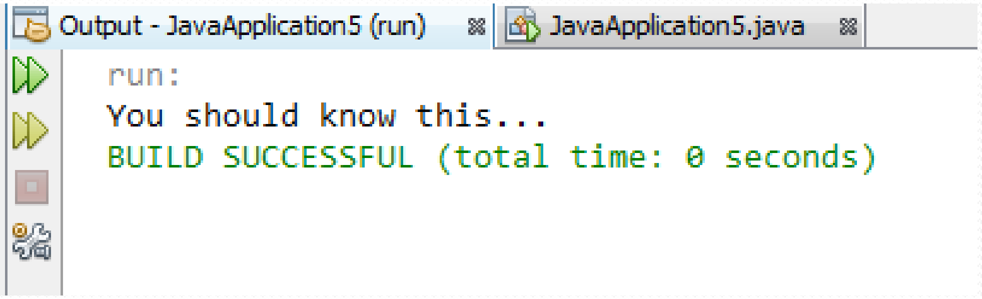
```
JOptionPane.showMessageDialog(null, "Hello world, JOptionPane");
```



Hello world, JOptionPane

- Console

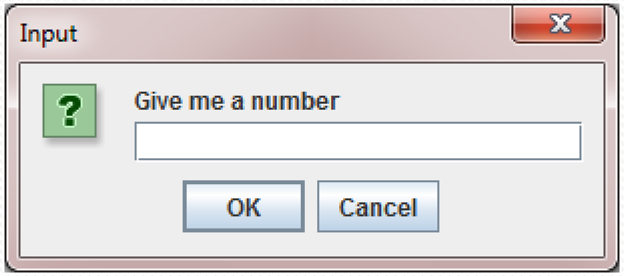
```
System.out.println("You should know this...");
```



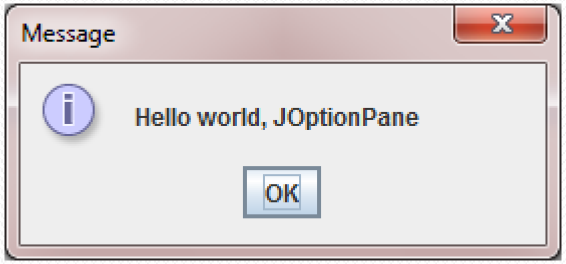
```
run:
You should know this...
BUILD SUCCESSFUL (total time: 0 seconds)
```

JOptionPane

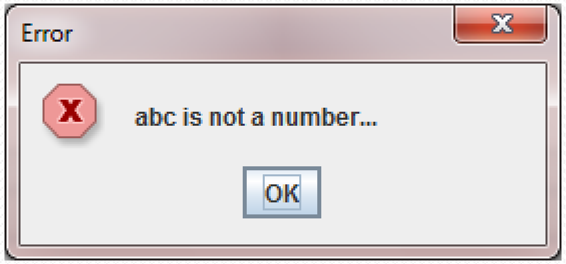
- JOptionPane : pop up a standard dialog box
  - prompts users for a value (input)
  - or informs them of something (output)



Give me a number



Hello world, JOptionPane



abc is not a number...

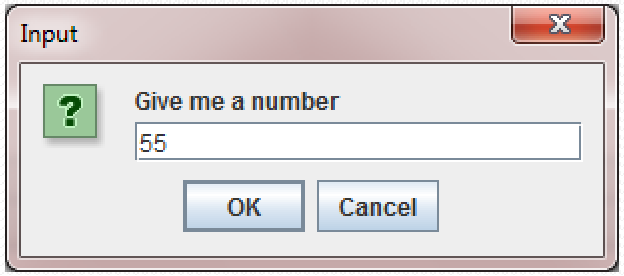
  - To use, put on head of source code

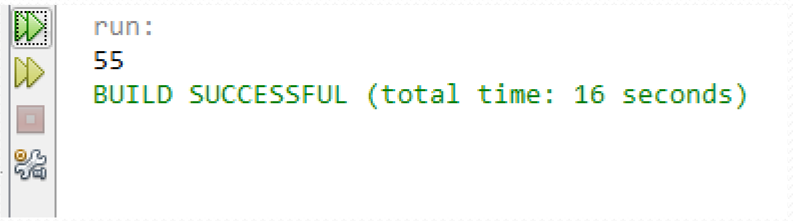
```
import javax.swing.JOptionPane;
```

JOptionPane -- Input

- `public static String showInputDialog(`
  `Component parentComponent, Object message)`
  - `parentComponent`: put `null`
  - `message`: your text
  - User's text is returned in showInputDialog()
- Example

```
String text = JOptionPane.showInputDialog(null, "Give me a number");
System.out.println(text);
```





```
run:
55
BUILD SUCCESSFUL (total time: 16 seconds)
```

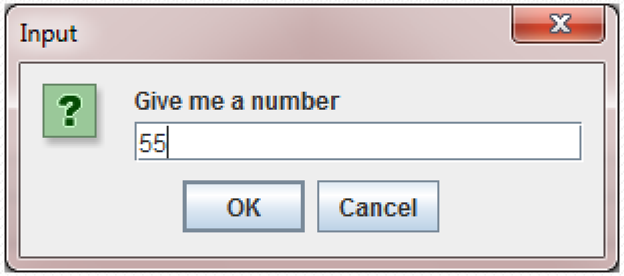
JOptionPane -- Parse input into number

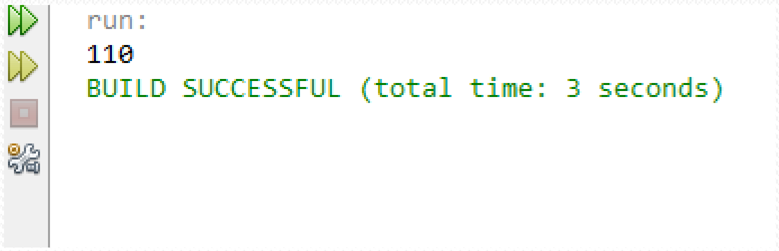
- `Integer.parseInt()`
  - Parse a string into integer

```
// put 55 into num
int num = Integer.parseInt("55");
```

- Parse a string from JOptionPane into integer

```
String text = JOptionPane.showInputDialog(null,
          "Give me a number");
int num = Integer.parseInt(text);
System.out.println(num*2);
```



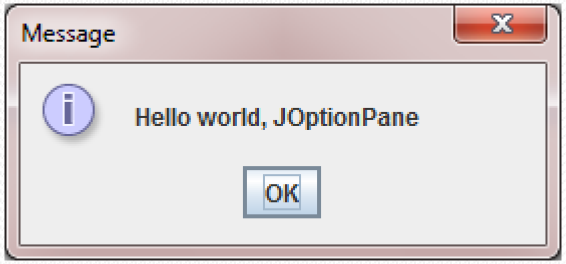


```
run:
110
BUILD SUCCESSFUL (total time: 3 seconds)
```

JOptionPane -- Output

- `public static void showMessageDialog(`
  `Component parentComponent, Object message)`
  - `parentComponent`: put `null`
  - `message`: your text
- Example:

```
JOptionPane.showMessageDialog(null, "Hello world, JOptionPane");
```

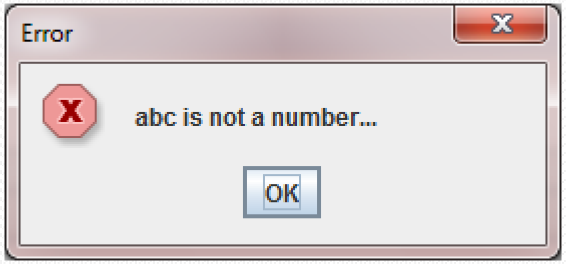


Hello world, JOptionPane

JOptionPane -- Output

- `public static void showMessageDialog(`
  `Component parentComponent, Object message,`
  `String title, int messageType)`
  - `title`: Dialog box title
  - `messageType`: Type of message to be display
    - (ERROR_MESSAGE, INFORMATION_MESSAGE, WARNING_MESSAGE, QUESTION_MESSAGE, or PLAIN_MESSAGE)
- Method overloading
- Example:

```
JOptionPane.showMessageDialog(null,
        "abc is not a number...",
        "Error",
        JOptionPane.ERROR_MESSAGE);
```



abc is not a number...

Summary

- Scanner class – read values from keyboard
- Java API Specification – search predefined class and methods
- Using Math class in calculations – help do calculations
- JOptionPane - to create dialogues (Graphical User Interface/ GUI.)

The end

Thank you!