Tutorial 4

Java

Topics

- Object creation
- Sending messages / method invocation
- JOptionPane.showInputDialog()
- Scanner
- If examples
- Math API
- Assignment 2

Object creation

- Declaring a variable to refer to an object

  Scanner keyboard = *null*; // refers to nothing yet

- Instantiating a class
  - The *new* keyword is a Java operator that ***creates*** an object.
- Initializing an object
  - The *new* operator is followed by a ClassName.

  Scanner keyboard = new Scanner(System.in);

Object creation

```
public class Point {
    public int x = 0;  // fields
    public int y = 0;

    // constructor, a special method for initializing a new object
    public Point(int a, int b) {
        x = a;  // copies the argument/ local variable a TO object field x
        y = b;  // copies the argument/ local variable b TO object field y
    }
}
```

How to create an object of this class?

Point originOne = new Point(23, 94); // a ← 23, b ← 94

Sending messages / method invocation

- Firstly, declare an object variable:

```
ClassName   objectVar;
```

- Then, create an object and store it:

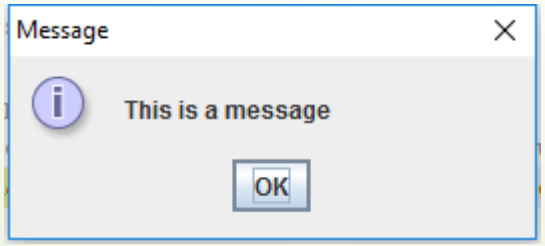
```
objectVar = new ClassName();
```

- Send a message as follows:

```
returnValue = objectVar.methodName( parameter );
```

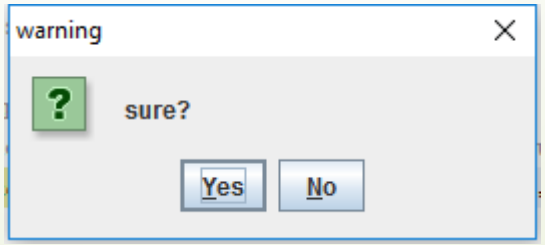
JOptionPane.showMessageDialog()

- showMessageDialog(Component parentComponent, Object message)
- Tell the user about something that has happened.
- JOptionPane.showMessageDialog(null,"This is a message");



JOptionPane.showConfirmDialog

- showConfirmDialog(Component parentComponent, Object message, String title, int optionType)
- Asks a confirming question, like yes/no/cancel.
- JOptionPane.showConfirmDialog(null,"sure?","warning",JOptionPane.YES_NO_OPTION);



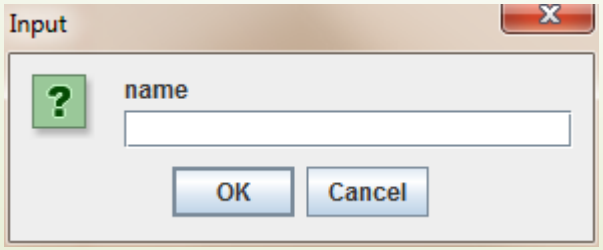
JOptionPane.showInputDialog()

public static String showInputDialog(Object message)

Shows a question-message dialog requesting input from the user.

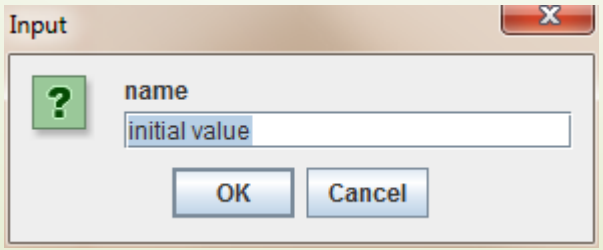
The return value is the string input by user.

```
String username = JOptionPane.showInputDialog("name");
```



JOptionPane.showInputDialog()

- public static String showInputDialog(Object message,Object initialSelectionValue)
- Shows a question-message dialog requesting input from the user, with the input value initialized to initialSelectionValue.
- The return value is the string input by user.
- JOptionPane.showInputDialog("name", "initial value");



Scanner

- You can use the Scanner next*() methods to get input from user through console.
- First, create a scanner object:

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

System.in is the "standard" input stream source.

Typically this stream corresponds to keyboard input.

Scanner

- Scanner next*() methods

double	**nextDouble**() Scans the next token of the input as a double.
float	**nextFloat**() Scans the next token of the input as a float.
int	**nextInt**() Scans the next token of the input as an int.
String	**nextLine**() Advances this scanner past the current line and returns the input that was skipped.
long	**nextLong**() Scans the next token of the input as a long.
short	**nextShort**() Scans the next token of the input as a short.

```
Scanner keyboard = new Scanner(System.in);
double d = keyboard.nextDouble();
```

This will get the console input from user, a double value.

What will happen if I input a string rather than a number?

Simple if-else examples

- The if-else Statement

```
void applyBrakes() {
  // the "if" clause: bicycle must be moving
   if (isMoving) {
     // the "true-case" clause: decrease current speed
      currentSpeed--;
   }
   else {
      System.err.println("The bicycle has already stopped!");
   }
}
```

Full example

- The following program, an if-else demo, assigns a grade based on the value of a test score, input from the user:
  - an A for a score of 90% or above;
  - a B for a score of 80% or above; and so on.
- A sample output from the program is:

  Input test score: 76

  Grade = C

```
class IfElseDemo {
    public static void main(String[] args) {
        System.out.print("Input test score: ");
        Scanner keyboard = new Scanner(System.in);
        int testScore = keyboard.nextInt();
        char grade;
        if (testScore >= 90) {
            grade = 'A';
        } else if (testScore >= 80) {
            grade = 'B';
        } else if (testScore >= 70) {
            grade = 'C';
        } else if (testScore >= 60) {
            grade = 'D';
        } else {
            grade = 'F';
        }
        System.out.println("Grade = " + grade);
    }
}
```

Math function APIs

- Remember to use the JAVA API Doc

Some functions that could be used in assignment 2

First import corresponding packages. Remember how to fix imports by NetBeans?

static double	**pow**(double a, double b) Returns the value of the first argument raised to the power of the second argument, a is the base and b is the exponent.
static double	**sqrt**(double a)Returns the correctly rounded positive square root of a double value.

```
double a = pow(2, 3);
double b = sqrt(4);
System.out.println(Math.round(3.14159));
```

Assignment 2

Aims:

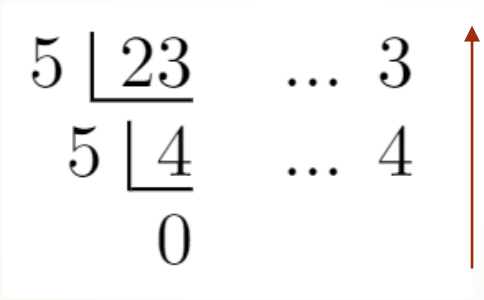
- To solve the number base conversion problem by writing a Java program.
- Practice using variables, expression and looping/branching statements

Assignment 2

- A decimal number sequence:
  - $123456_{10} = 1*10^5 + 2*10^4 + 3*10^3 + 4*10^2 + 5*10^1 + 6*10^0$
- In this assignment, we are going to deal with numbers in base 5 and base 7
- In general, one may use the following formula to convert a base N_{10} number to base 10 (N itself is represented in base 10):
  - $abcdef_N = a*N^5 + b*N^4 + c*N^3 + d*N^2 + e*N^1 + f*N^0$
  - For example, for N = 5, then
    - $123403_5 = 1*5^5 + 2*5^4 + 3*5^3 + 4*5^2 + 0*5^1 + 3*5^0 = 4853_{10}$

Assignment 2

- To reverse the process, another algorithm, known as repeated division, is required
- For example, we can change a base 10 number, say 23_{10}, to base 5 by



- We take the remainders from bottom to top. The answer is 43_5
- To change a base 10 number to any other bases, say N, we are required to perform a similar procedure by replace 5 with N. In our assignment, N can only be 5 or 7.

Assignment 2

- In this assignment, you are required to write a program to perform number conversion among base 5, 7 and 10 numbers.
- You are required to obtain the input number from the console standard input using Scanner and System.in.
- A sample run of the program is shown below:

```
Enter a base 10 number: 123456
The number in base 5 and 7 are: 12422311, 1022634
Enter a base 5 number: 123403
The number in base 7 and 10 are: 20102, 4853
Enter a base 7 number: 123456
The number in base 5 and 10 are: 1213000, 22875
```

Assignment 2

- There are **at most six digits** in the any of the input numbers.
- You are **NOT** required to validate the input numbers. It means that you can assume all input numbers are valid.
- In the conversion process, you are **NOT** allowed to use any elements/methods related to the class String.

Assignment 2

- You shall define one package **exercise**
- and one class called **BaseConversion**
- in a new NetBeans project named **BaseConversion**
- The class shall contain a main method that performs all the operations including input, conversion and output.
- Optionally, you may define more than one method in addition to the main method to carry out the conversion.

Notes

- Example for Conversion: $1234_5 \rightarrow 365_7$
- Step1:
  - $1234_5 = (1*5^3 + 2*5^2 + 3*5^1 + 4*5^0)_{10} = 194_{10}$
- Step2:

$$\begin{array}{rll} 7 \,|\, 194 & \ldots & 5 \\ 7 \,|\, 27 & \ldots & 6 \\ 7 \,|\, 3 & \ldots & 3 \\ 0 & & \end{array}$$

- Answer: 365_7

Notes

- Integer division/modulus
  - 194 / 7 = 27
  - 194 % 7 = 5
- Self-study Exercises: Reversing the digits of an integer

Input: 1234
Output: 4321
1234 + 4321 = 5555

Input: 131
Output: 131
131 + 131 = 262

End