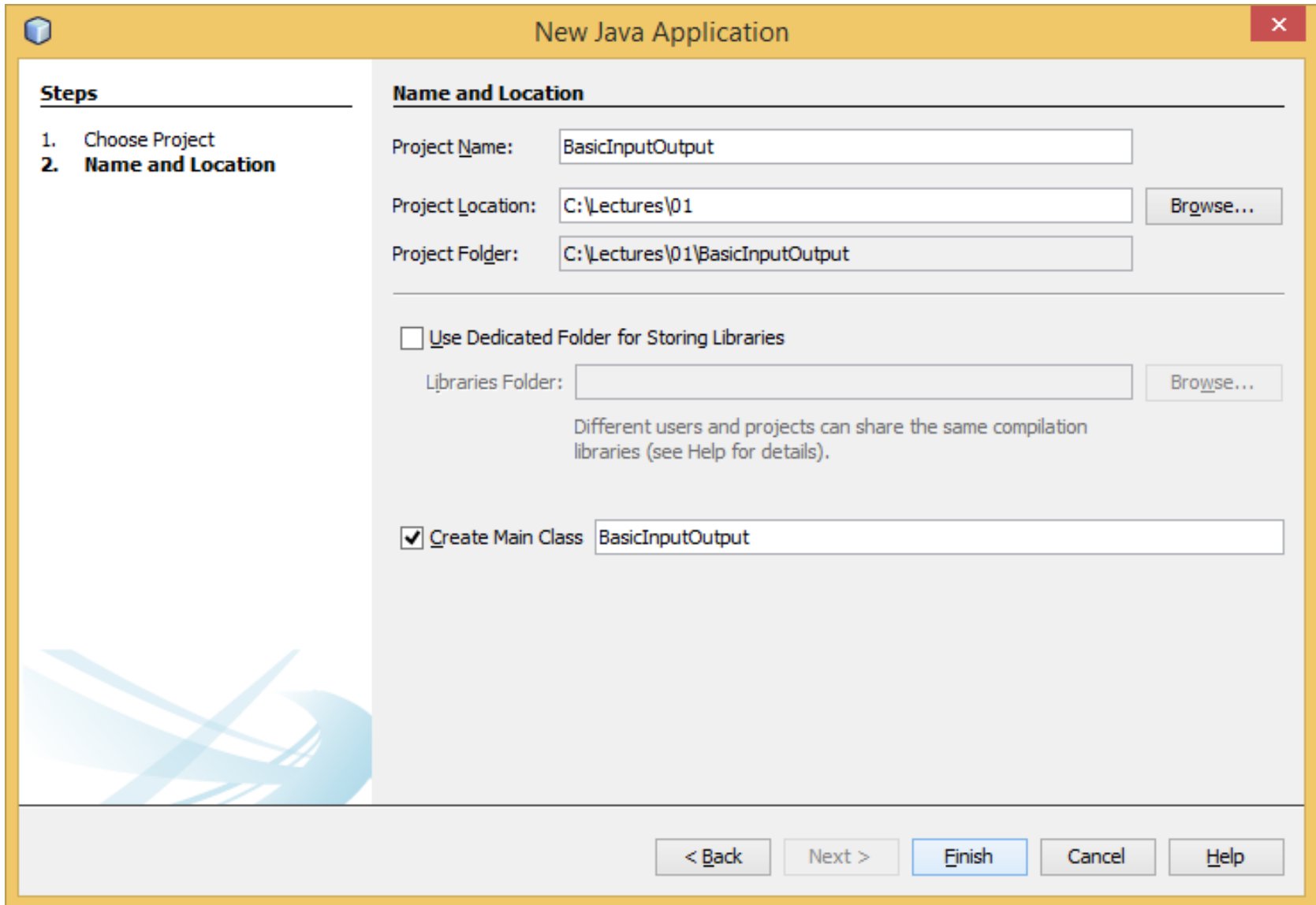


Εισαγωγή στη Java

Εκτύπωση τιμών στην οθόνη



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: BasicInputOutput

Project Location: C:\Lectures\01 Browse...

Project Folder: C:\Lectures\01\BasicInputOutput

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class BasicInputOutput

< Back Next > **Finish** Cancel Help

To namespace System

The screenshot shows an IDE window for the `java.lang.System` class. The main pane displays the class's documentation, including a description of the `err` stream and a list of methods. The left pane shows the class hierarchy, and the right pane shows the class's methods.

Class Documentation:

```
public static final PrintStream err
```

The "standard" error output stream. This stream is already open and ready to accept output data.

Typically this stream corresponds to display output or another output destination specified by the host environment or user. By convention, this output stream is used to display error messages or other information that should come to the immediate attention of a user even if the principal output stream, the value of the variable `out`, has been redirected to a file or other destination that is typically not continuously monitored.

Methods:

Method	Return Type
<code>err</code>	<code>PrintStream</code>
<code>in</code>	<code>InputStream</code>
<code>out</code>	<code>PrintStream</code>
<code>arraycopy(Object src, int srcPos, Object dest, int destPos, int length)</code>	<code>void</code>
<code>clearProperty(String key)</code>	<code>String</code>
<code>console()</code>	<code>Console</code>
<code>currentTimeMillis()</code>	<code>long</code>
<code>exit(int status)</code>	<code>void</code>
<code>gc()</code>	<code>void</code>
<code>getProperties()</code>	<code>Properties</code>
<code>getProperty(String key)</code>	<code>String</code>
<code>getProperty(String key, String def)</code>	<code>String</code>
<code>getSecurityManager()</code>	<code>SecurityManager</code>
<code>getenv()</code>	<code>Map<String, String></code>
<code>getenv(String name)</code>	<code>String</code>
<code>identityHashCode(Object x)</code>	<code>int</code>
<code>inheritedChannel()</code>	<code>Channel</code>

To namespace System.out

The screenshot shows an IDE window titled "BasicInputOutput". The left pane displays the source code of the `BasicInputOutput` class. The right pane shows the documentation for `java.io.PrintStream`, specifically the `append(CharSequence csq)` method. The bottom pane shows the output of the program, which includes the text "BUILD SUCCESSFUL (total time: 0s)".

Source Code (BasicInputOutput.java):

```
11 public class BasicInputOutput {
12     /**
13      * @param args
14      */
15     public static void main(String[] args) {
16         // TODO: write your code here
17         System.out.println("Hello, World!");
18         System.out.println("Hello, World!");
19     }
20 }
21
22
23
```

Documentation (java.io.PrintStream):

`public PrintStream append(CharSequence csq)`

Appends the specified character sequence to this output stream.

An invocation of this method of the form `out.append(csq)` behaves in exactly the same way as the invocation

```
out.print(csq.toString())
```

Depending on the specification of `toString` for the character sequence `csq`, the entire sequence may not be appended. For instance, invoking then `toString` method of a

Method List:

- `print(Object obj)` void
- `print(String s)` void
- `print(boolean b)` void
- `print(char c)` void
- `print(char[] s)` void
- `print(double d)` void
- `print(float f)` void
- `print(int i)` void
- `print(long l)` void
- `printf(String format, Object... args)` PrintStream
- `printf(Locale l, String format, Object... args)` PrintStream
- `println()` void
- `println(Object x)` void
- `println(String x)` void
- `println(boolean x)` void
- `println(char x)` void
- `println(char[] x)` void

Output - BasicInputOutput (run):

```
run:
Πληκτρολόγησε έναν ακέραιο
BUILD SUCCESSFUL (total time: 0s)
```

Η λειτουργία println

The screenshot shows an IDE window titled "BasicInputOutput" with the "Source" tab selected. The code editor displays a Java class with a static method that calls `System.out.println()`. A tooltip is visible over the `println()` call, showing the `java.io.PrintStream` documentation. The documentation includes the signature `public void println(Object x)`, a description of the method's behavior, and a parameter list. Below the tooltip, a list of `println` overloads is shown, with `println(Object x)` selected. The bottom of the IDE shows the "Output" window with the message "BUILD SUCCESSFUL (total time: 0s)".

java.io.PrintStream

`public void println(Object x)`

Prints an Object and then terminate the line. This method calls at first `String.valueOf(x)` to get the printed object's string value, then behaves as though it invokes [PrintStream.print\(String\)](#) and then [PrintStream.println\(\)](#).

Parameters:

- `x` - The Object to be printed.

Overloads:

- `println()` void
- `println(Object x)` void**
- `println(String x)` void
- `println(boolean x)` void
- `println(char x)` void
- `println(char[] x)` void
- `println(double x)` void
- `println(float x)` void
- `println(int x)` void
- `println(long x)` void
- `toString()` String
- `wait()` void
- `wait(long timeout)` void
- `wait(long timeout, int nanos)` void
- `write(byte[] b)` void
- `write(int b)` void
- `write(byte[] buf, int off, int len)` void

Output - BasicInputOutput (run)

run: Πληκτρολόγησε έναν ακέραιο αριθμό
BUILD SUCCESSFUL (total time: 0s)

Ο πηγαίος κώδικας της εφαρμογής

```
7  /**
8   *
9   * @author Nick Z. Zacharis
10  */
11  public class BasicInputOutput {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          // TODO code application logic here
18          System.out.println("Πληκτρολόγησε έναν ακέραιο αριθμό : ");
19      }
20
21  }
22
```

Output - BasicInputOutput (run) ×

```
run:
Πληκτρολόγησε έναν ακέραιο αριθμό :
BUILD SUCCESSFUL (total time: 0 seconds)
```

Το αποτέλεσμα της εκτέλεσης
του προγράμματος

Χρήση της λειτουργίας print και println

```
11 public class BasicInputOutput {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         // TODO code application logic here
18         System.out.print("Το αποτέλεσμα 2 + 2 = ");
19         System.out.println(2+2);
20         System.out.println("Το αποτέλεσμα 3 + 3 = " + (3 + 3));
21     }
22 }
23
```

Output - BasicInputOutput (run) X

```
run:
To αποτέλεσμα 2 + 2 = 4
To αποτέλεσμα 3 + 3 = 6
BUILD SUCCESSFUL (total time: 0 seconds)
```

Run...

Ανάγνωση τιμών από το πληκτρολόγιο

```
public class BasicInputOutput {  
  
    public static void main(String[] args) {  
  
        System.out.print("Type in console : ");  
        String message = System.console().readLine();  
        System.out.println("Console Message : " + message);  
  
        System.out.print("Type in scanner : ");  
        message = new java.util.Scanner(System.in).next();  
        System.out.println("Scanner Message : " + message);  
  
        message = javax.swing.JOptionPane.showInputDialog("Πληκτρολόγησε μια τιμή :");  
        System.out.println("ShowInputDialog : " + message);  
    }  
}
```

Run...

Βασικοί τύποι δεδομένων και μεταβλητές

Βασικοί Τύποι Δεδομένων

Τύπος	bit	Εύρος	Προκαθορισμένη Τιμή
byte	8	-128..127	0
short	16	-32768..32767	0
int	32	-2,147,483,648.. 2,147,483,647	0
long	64	-9,223,372,036,854,775,808 9,223,372,036,854,775,807	0
float	32	1.40129846432481707e-45 3.40282346638528860e+38	0.0f
double	64	4.94065645841246544e-324d 1.79769313486231570e+308d	0.0d
boolean	1	true ή false	false
char	16	Unicode χαρακτήρας	'\u0000'

Μεταβλητές

```
int  a;           // Δημιουργία μιας ακεραίας μεταβλητής
int k, m;         // Δημιουργία δύο ακεραίων μεταβλητών
int d = 12, e, z = 5; // Δημιουργία και αρχικοποίηση ακεραίων μεταβλητών
byte f = 32;      // Δημιουργία και αρχικοποίηση byte μεταβλητής
double pi = 3.141; // Δημιουργία και αρχικοποίηση πραγματικής μεταβλητής
float fpa = 23.5f // Δημιουργία και αρχικοποίηση πραγματικής μεταβλητής
char x = 't';     // Δημιουργία και αρχικοποίηση μεταβλητής χαρακτήρα
```

Εφαρμογή με μεταβλητές

```
public class DataTypes {  
  
    public static void main(String[] args) {  
        float x = 23.5f;  
        float k = 12.56f;  
  
        System.out.println( "x * k = " + (x * k) );  
  
        System.out.print( "Type m : ");  
        float m = new java.util.Scanner(System.in).nextFloat();  
        System.out.print( "Type n : ");  
        float n = new java.util.Scanner(System.in).nextFloat();  
  
        System.out.println( "m * n = " + (m * n) );  
    }  
}
```

Αριθμητικοί Τελεστές

Θεωρούμε ότι η μεταβλητή A έχει τιμή 24 ενώ το B έχει τιμή 4

Τελεστής	Περιγραφή	Παράδειγμα
+	Πρόσθεση δύο τιμών	$A + B$ will give 28
-	Αφαίρεση της δεξιάς από την αριστερή τιμή	$A - B$ will give 20
*	Γινόμενο δύο τιμών	$A * B$ will give 96
/	Διαίρεση της αριστερής από τη δεξιά τιμή	A / B will give 6
%	Υπόλοιπο της διαίρεσης της αριστερής από τη δεξιά τιμή	$B \% A$ will give 0
++	Αύξηση κατά 1	$A++$ gives 25
--	Μείωση κατά 1	$B--$ gives 3

Όλες οι πράξεις έχουν προτεραιότητα εκτέλεσης ώστε να υπολογίζονται σύνθετα αποτελέσματα. Η παρένθεση έχει τη μεγαλύτερη προτεραιότητα.

Άσκηση στους τύπους δεδομένους

Δημιουργείστε μια εφαρμογή για τον υπολογισμό της μισθοδοσίας μιας εταιρίας. Το πρόγραμμα θα διαβάζει :

τις ημέρες εργασίας, (π.χ. 10)

το μεροκάματο (π.χ. 50)

θα υπολογίζει και θα εκτυπώνει την ασφάλεια του, η οποία υπολογίζεται σαν 7 % του μισθού, $(10 * 50 * 7 / 100)$

το φόρο του, υπολογίζεται σαν 5 % του μισθού, $(10 * 50 * 5 / 100)$

καθώς και το καθαρό ποσό της αμοιβής. $(10 * 50 - \text{φόρος} - \text{ασφάλεια})$

Άσκηση στους τύπους δεδομένους

```
public class Company {  
    public static void main(String[] args) {  
        int days;  
        float pay, money, total, foros, asfaleia;  
  
        System.out.print("Type days : ");  
        days = new java.util.Scanner(System.in).nextInt();  
        System.out.print("Type days pay : ");  
        pay = new java.util.Scanner(System.in).nextFloat();  
        money = days * pay;  
        foros = (money * 0.05f);  
        asfaleia = (money * 0.07f);  
        System.out.println("Tax = " + foros);  
        System.out.println("Social security = " + asfaleia);  
        System.out.println("Salary = " + (money - asfaleia - foros));  
    }  
}
```

Run...

Άσκηση

Δημιουργείστε μια Java εφαρμογή για το υπολογισμό του όγκου ενός κώνου

$$V = \frac{1}{3} \cdot \pi \cdot r^2 \cdot h$$

