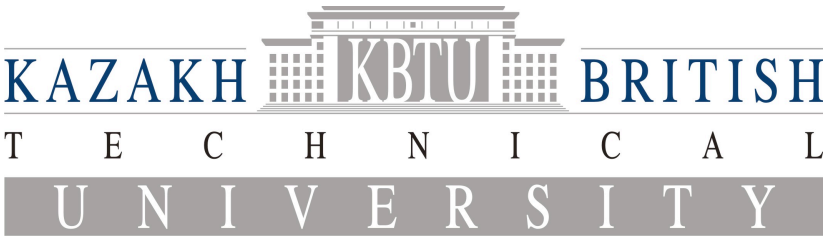


Faculty of Information Technology

Department of Computer Engineering

Object-Oriented programming

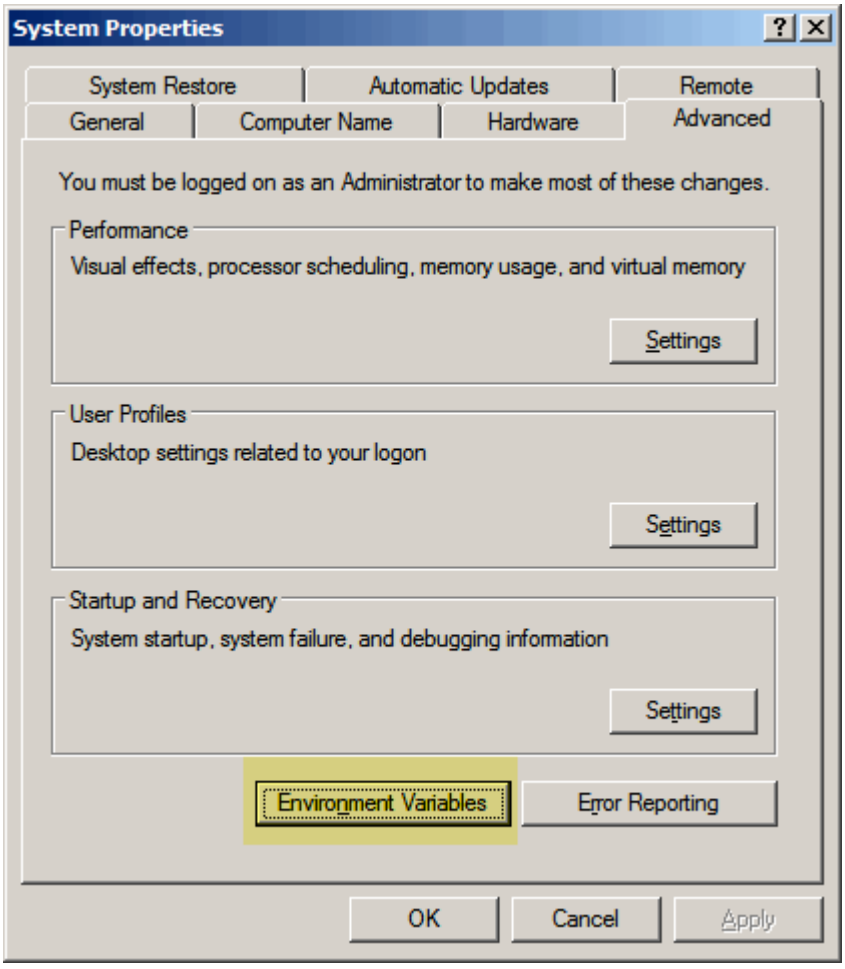
Methodological instructions for laboratory works

Compiled by:

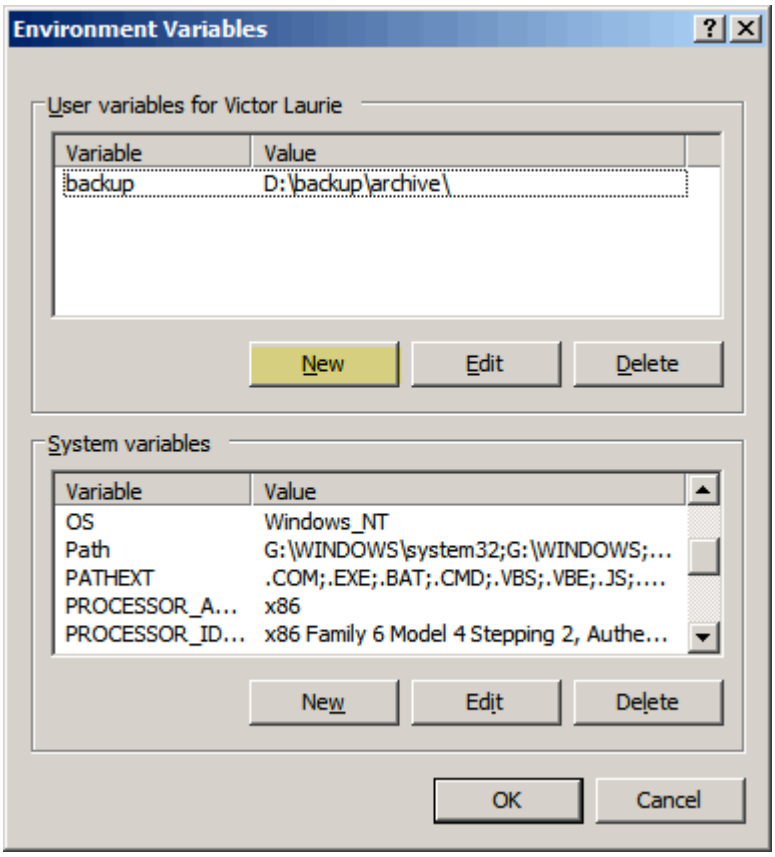
P. Shamoi

Almaty 2017

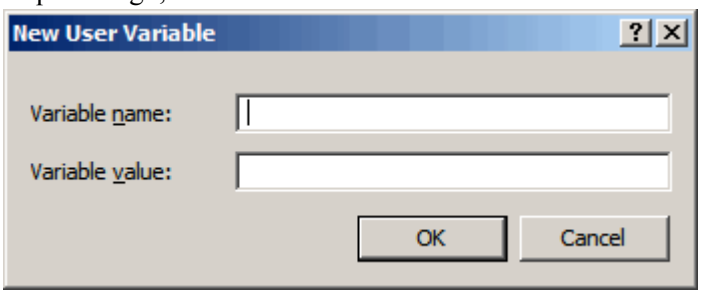
1. **Setting up your programming environment.**
   a. Download and save JDK and eclipse
   b. Run JDK exe file and follow installation steps.
   c. To make available the execution of java compiler you should set your *path* variable. But this step is not necessary if you are going to program in Eclipse, as I highly recommend to you. If you are going to use notepad, Emeditor, etc., do this. The *path* to a file is basically its address on the computer. It tells programs how to find a file. It is the drive plus any directories and sub-directories where the file is located. The %PATH% environment variable specifies the command search path. Typically, this is a group of directories where executable files that are repeatedly used are to be found. Examples of the default values are listed in the first table above; in this case, they are the Windows directory and two of its important system sub-directories. To see what is in the PATH variable on a computer, open a command window and enter "echo %PATH%". A more permanent way to manage environment variables is provided in the System Properties dialog box. Open **Control Panel-Performance and Maintenance-System** (or right-click on **My Computer** and choose "Properties"). In the box that opens, click the "Advanced" tab to obtain the dialog box shown below. Next, click the button "Environment Variables".



The figure below shows the "Environment Variables" dialog box that opens next. It lists two kinds of variable- those that apply only to the current user and those that apply to the whole system.



The box for adding a new user variable is shown below. Generally, this is likely to be a directory that you use frequently but can be any string of less than 8192 bytes. The maximum total size for all environment variables, including variable names and the "equals" sign, is 32767 characters.



Type *path* in Variable name field and copy path to bin directory of java compiler to Variable value field. Click Ok.

d. Using Eclipse.

1. Start Eclipse.

2. Create a new Java Project:
   a. **File->New->Project**.
   b. Select "**Java**" in the category list.
   c. Select "**Java Project**" in the project list. Click "**Next**".
   d. Enter a project name into the **Project name** field, for example, "`Hello World Project`".
   e. Click "**Finish**"--It will ask you if you want the Java perspective to open. (You do.)
3. Create a new Java class:
   a. Right click on your project ad choose New-> Class.
   b. Enter "`HelloWorld`" into the **Name** field.

c. Click the checkbox indicating that you would like Eclipse to create a "`public static void main(String[] args)`" method.
d. Click "**Finish**".

4. A Java editor for `HelloWorld.java` will open. In the main method enter the following line.
```
System.out.println("Hello World");
```
5. Save using **ctrl-s**. This automatically compiles `HelloWorld.java`.
6. Click the "**Run**" button in the toolbar (looks like a little man running).
7. You will be prompted to create a Launch configuration. Select "**Java Application**" and click "**New**".
8. Click "**Run**" to run the Hello World program. The console will open and display "**`Hello World`**".

e. If you are not using eclipse…

When you save your public class declaration in a file, the file name must be the class name followed by the "**.java**" file-name extension. For our application, the file name is Welcome1.java.

1. To compile the program, type in a command line.
```
javac Welcome1.java
```

2. If the program contains no syntax errors, the preceding command creates a new file called Welcome1.class (known as the **class file** for Welcome1) containing the Java bytecodes that represent our application.

3. When we use the java command to execute the application, these bytecodes will be executed by the JVM.
```
java Welcome1
```