INF160
IS Development Environments

AUBG, COS dept

Reference books:

Baltzan Paige, Business Driven Information Systems, McGraw-Hill/Irwin, 3e, 2012.

Doar Matthew B., Practical Development Environments, O'Reilly, 2005.

Any C++, C#, Java, VBasic book available in AUBG library

Course lecturer: Assoc. Prof. Stoyan Bonev, PhD

INF160 IS Development Environments
AUBG, COS dept, Fall semester 2011

Lecture 06

Title:

Dev Env: NetBeans

(Extract from Syllabus)

Reference: www.netbeans.org

Lecture Contents:

- NetBeans – introduction
- NetBeans – functionality
  - NetBeans IDE Java Quick Start Tutorial (http://netbeans.org/kb/docs/java/quickstart.html)
  - Introduction to GUI Building (http://netbeans.org/kb/docs/java/gui-functionality.html)
- NetBeans – configuration

NetBeans – introduction

- NetBeans refers to both
  - NetBeans Platform
  - NetBeans IDE
- NetBeans Platform is a reusable framework for simplifying the development of Java Swing desktop applications.
- NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of all Java application types

NetBeans – introduction

- NetBeans IDE 7.0
- Develop desktop, mobile and web applications with Java, PHP, C/C++ and more.

- Runs on Windows, Linux, Mac OS X and Solaris. NetBeans IDE is open-source and free

NetBeans – functionality
NetBeans IDE Java Quick Start

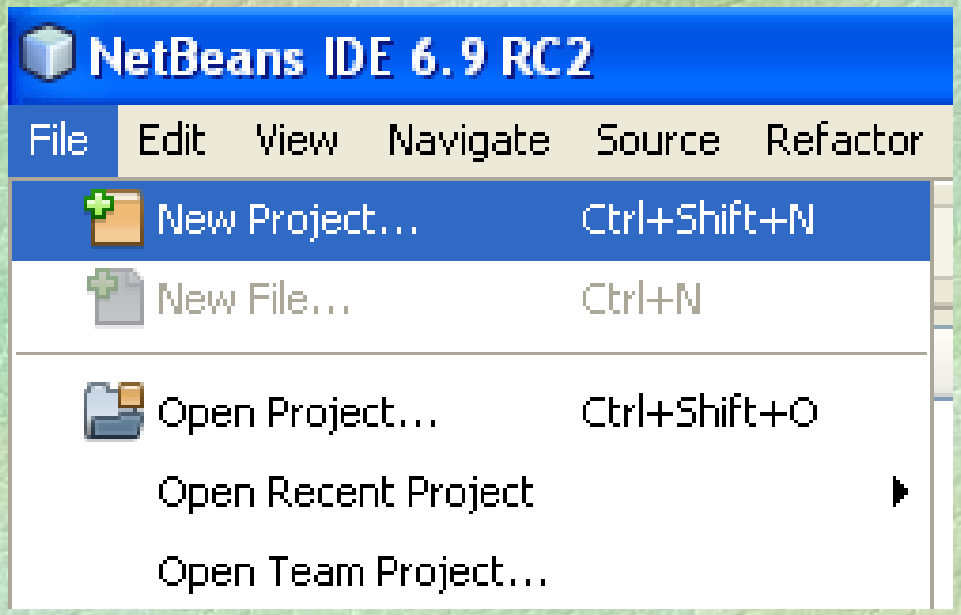
- Setting up the project
  - To Create an IDE project
- Adding Code to the Generated Source File
- Compiling and running a project
- Building and deploying the application (.jar file)
- Create javadoc
- Packaging and Distributing Java Desktop Applications
- Tips for NetBeans users

Setting Up the Project

- .

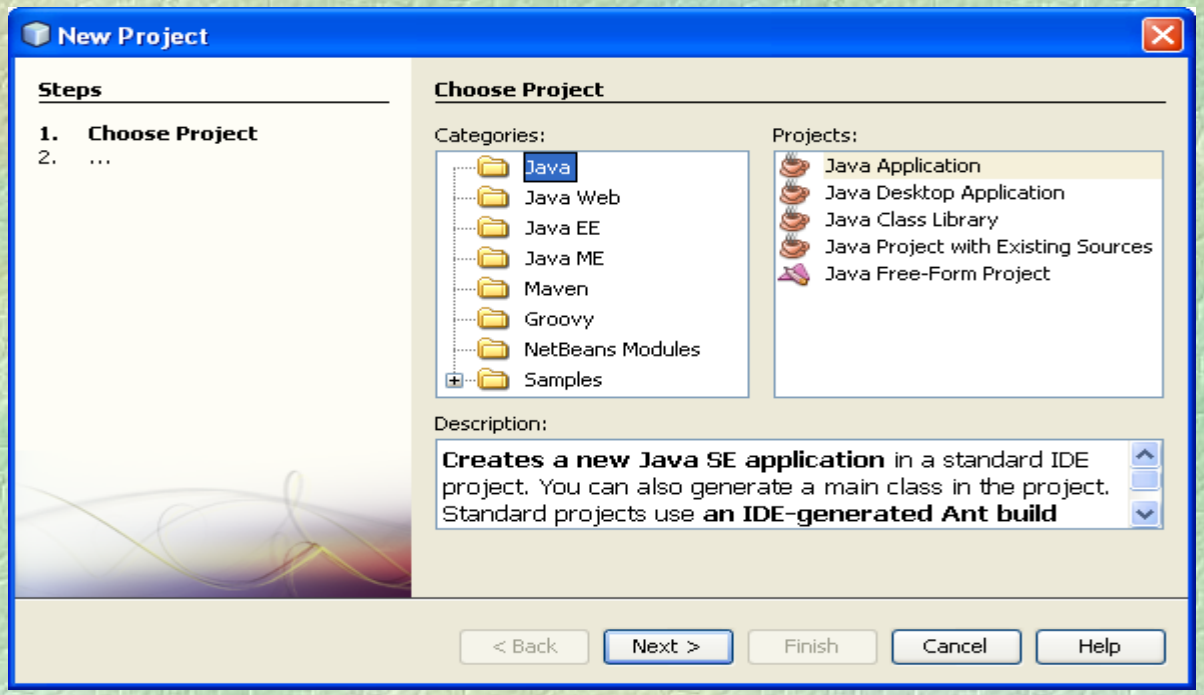
To create an IDE project:

- 1. Start NetBeans IDE.
- 2. In the IDE, choose File > New Project… (Ctrl-Shift-N), as shown in the figure below.

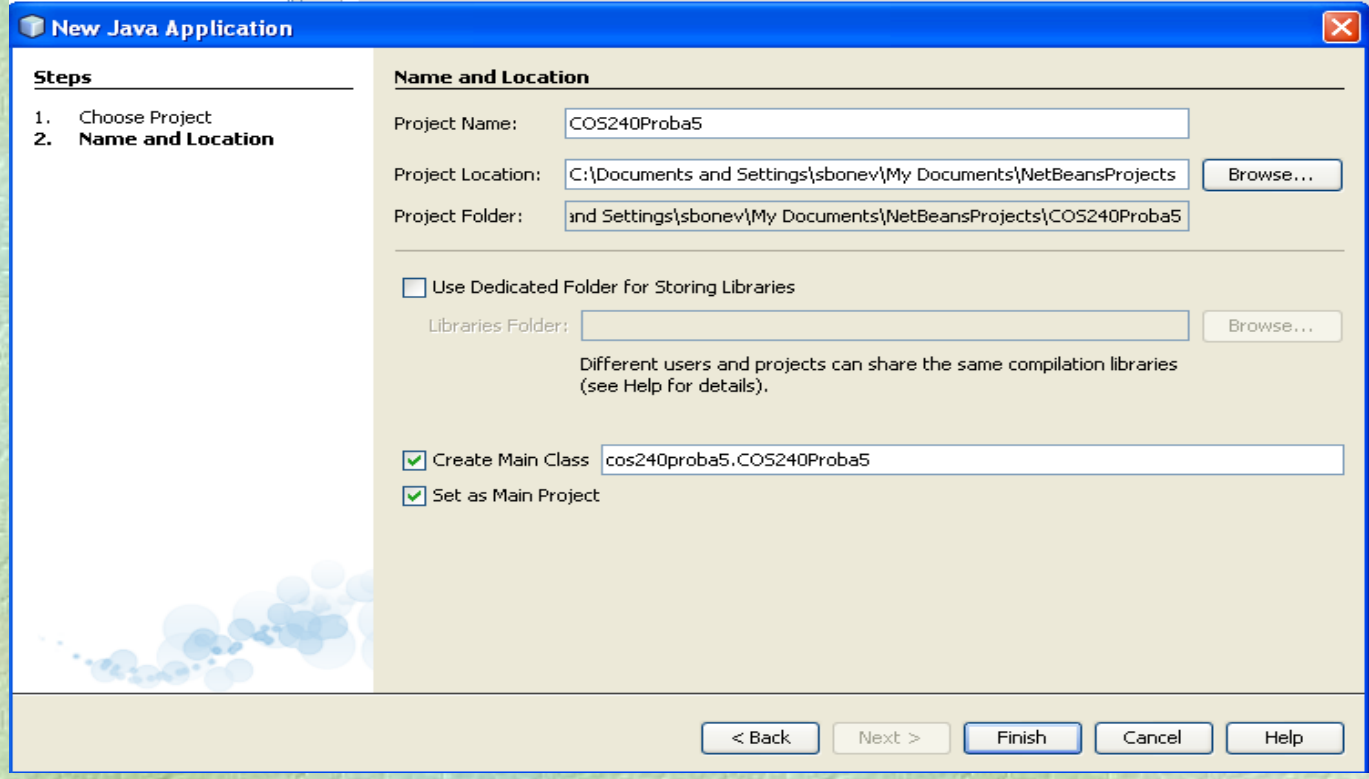


To create an IDE project:

- 3. In the New Project wizard, expand the Java category and select Java Application as shown in the figure below. Then click Next.

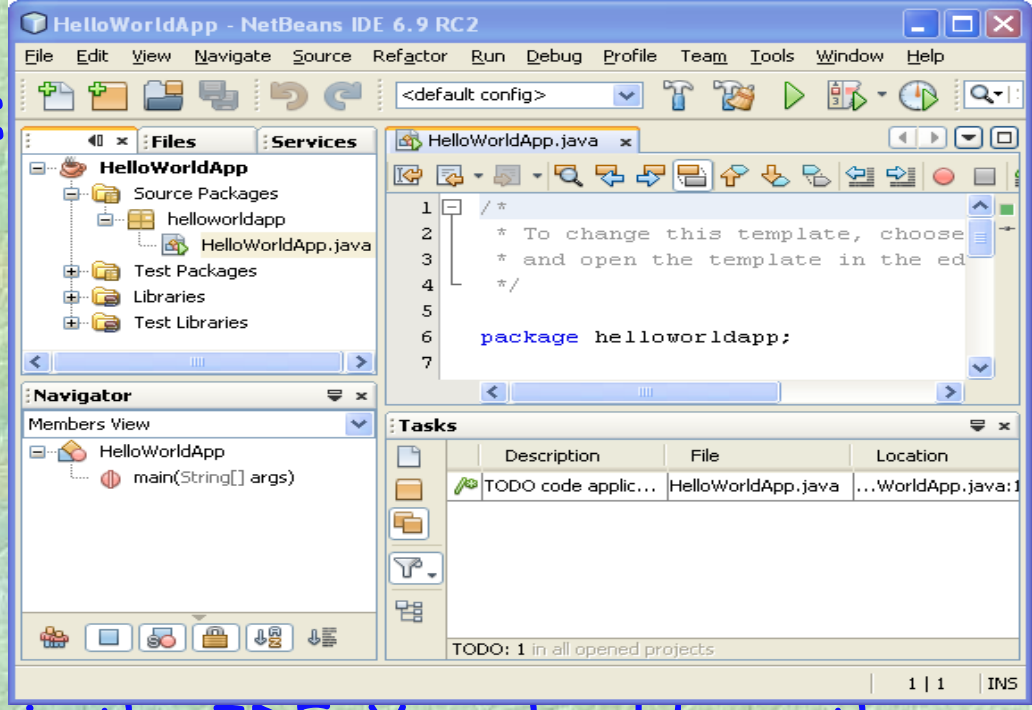


To create an IDE project:



- 4. In the Name and Location page of the wizard, do the following (as shown in the figure above):
  - In the Project Name field, type HelloWorldApp.
  - Leave the Use Dedicated Folder for Storing Libraries checkbox unselected.
  - In the Create Main Class field, type helloworldapp.HelloWorldApp (if it is empty).
  - Leave the Set as Main Project checkbox selected.

To create an IDE project:



- 5. Click Finish.

The project is created and opened in the IDE. You should see the following components:

- The Projects window, which contains a tree view of the components of the project, including source files, libraries that your code depends on, and so on.
- The Source Editor window with a file called HelloWorldApp open.
- The Navigator window, which you can use to quickly navigate between elements within the selected class.
- The Tasks window, which lists compilation errors as well other tasks that are marked with keywords.

Adding Code to the Generated Source File

- Because you have left the Create Main Class checkbox selected in the New Project wizard, the IDE has created a skeleton main class for you. You can add the "Hello World!" message to the skeleton code by replacing the line:

```
// TODO code application logic here
```

with the line: `System.out.println("Hello World!");`

- Save the change by choosing File > Save.
- The file should look something like the following code sample.

Adding Code to the Generated Source File

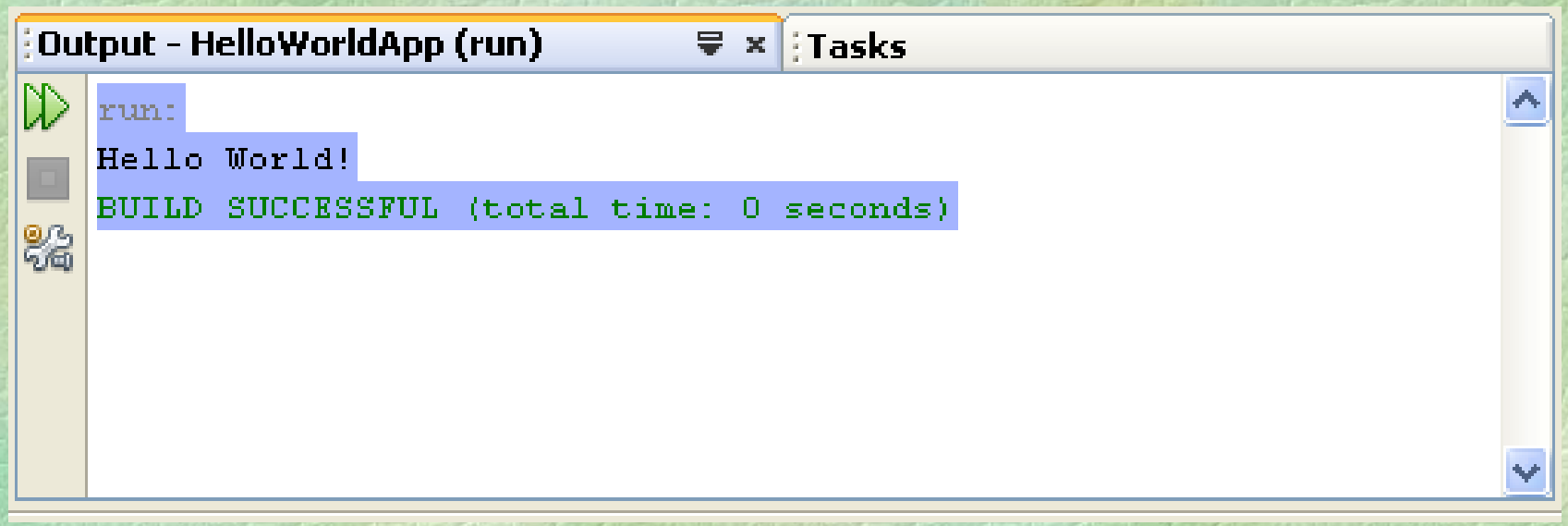
```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package helloworldapp;

/**
 *
 * @author <your name>
 */
public class HelloWorldApp {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
            System.out.println("Hello World!");
    }

}
```

Compiling and running a project

- Because of the IDE's Compile on Save feature, you do not have to manually compile your project in order to run it in the IDE. When you save a Java source file, the IDE automatically compiles it.

- To run the program:
  - Choose Run > Run Main Project (F6).
  - The next figure shows what you should now see.
  - Congratulations! Your program works.

Compiling and running a project



Compiling and running a project

- If there are compilation errors, they are marked with red glyphs in the left and right margins of the Source Editor.
  - The glyphs in the left margin indicate errors for the corresponding lines.
  - The glyphs in the right margin show all of the areas of the file that have errors, including errors in lines that are not visible.
- You can mouse over an error mark to get a description of the error.
- You can click a glyph in the right margin to jump to the line with the error.

Compiling and running a project

- Here are more options to process the project:
- To test the program:
  - Choose Run > Test Project(<prj name>) (Alt-F6)
  - Try the command to know what you should now see.
  - The next figure shows what you should now see.

Compiling and running a project

- Here are more options to process the project:
- To build manually (without run) the program:
  - Choose Run > Build Main Project (F11)
  - Try the command to know what you should now see.
  - The next figure shows what you should now see.

Compiling and running a project

- Here are more options to process the project:
- To duplicate the recent command given:
  - Choose Run > Repeat Build/Run (<prj name>)
  - Try the command to know what you should now see.
  - The next figure shows what you should now see.

Compiling and running a project

- Here are more options to process the project:
- To establish project settings:
  - Choose Run > Set project configuration >
    - <default setting>
    - Customize…
  - Try the command to know what you should now see.
  - The next figure shows what you should now see.

Compiling and running a project

- Here are more options to process the project:
- To establish Main project settings:
  - Choose Run > Set Main project >
    - None
    - Select among all projects for the current NetBeans session
  - Try the command to know what you should now see.
  - The next figure shows what you should now see.

Compiling and running a project

- Here are more options to process the project:
- To create archive (.jar) file for the program:
  - Choose Run > Clean and Build Main Project (Shift-F11)
  - Try the command to know what you should now see.
  - The next figure shows what you should now see.
  - For details see next slide.

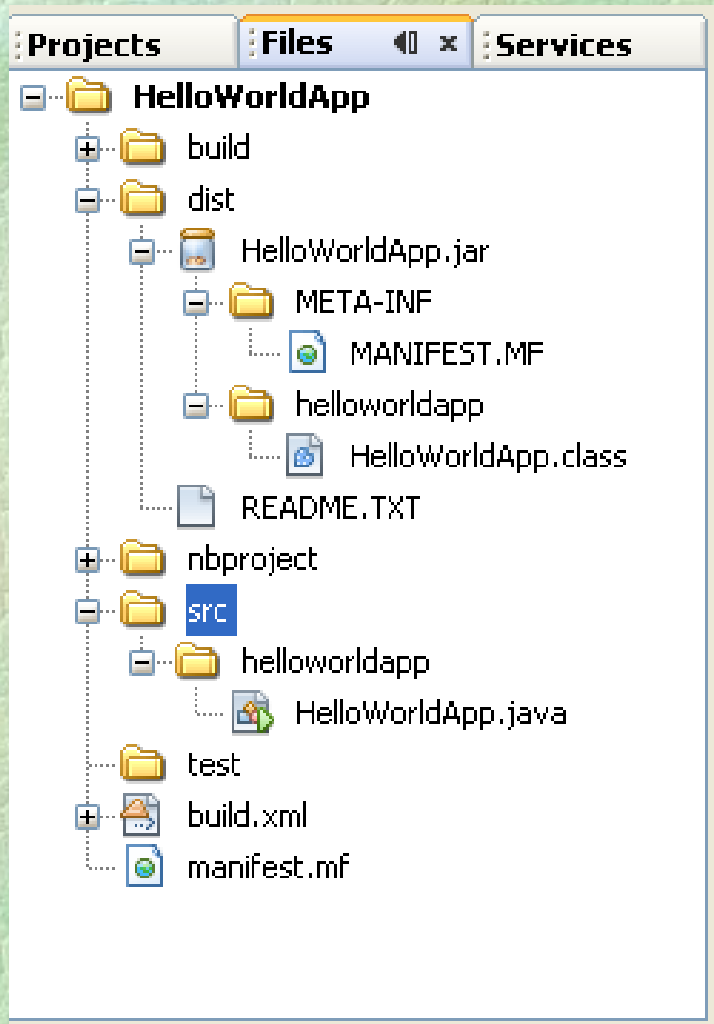
Building and deploying the application (.jar file)

- Once you have written and test run your application, you can use the Clean and Build command to build your application for deployment. When you use the Clean and Build command, the IDE runs a build script that performs the following tasks:
  - Deletes any previously compiled files and other build outputs.
  - Recompiles the application and builds a JAR file containing the compiled files.

- To build your application:
  - Choose Run > Clean and Build Main Project (Shift-F11)

- You can view the build outputs by opening the Files window and expanding the HelloWorldApp node. The compiled bytecode file HelloWorldApp.class is within the build/classes/helloworldapp subnode. A deployable JAR file that contains the HelloWorldApp.class is within the dist node.

Building and deploying the application (.jar file)

- 

Tips for NetBeans users

- The Compile on Save feature can be turned off in the Project Properties window. Right-click your project, select Properties. In the Properties window, choose the Compiling tab. The Compile on Save checkbox is right at the top.
- Note that in the Project Properties window you can configure numerous settings for your project: project libraries, packaging, building, running, etc.T

Thank You
For
Your Attention!