IDE Review

- BlueJ
- NetBeans
- Eclipse

Namespace, Package, Classpath

- `baseDir/x`
- `baseDir/x/y`
- `baseDir/x/y/z`
- If class X is defined as below, which directory should it be in?
- X.java
- `package x.y.z;`
- `public class X {}`

Java Archive

- eXtract
  - jar xf archivename
  - jar xf var-packer-assign.jar
  - Produces the folowing directory structure
  - `./VarPacker`
  - `./VarPacker/Lists`

Java Archive

- Create
  - jar cf new-archive-file path-to-archive
  - For the following structure:
  - `./VarPacker`
  - `./VarPacker/Lists`
  - jar cf var-packer-assign.jar ./VarPacker *txt
- Table of contents
  - jar tf archivename

Importing Files

- BlueJ
  - extract jar in new directory, say, Import
  - import from this new Import directory
  - automatically reads in all java files
- NetBeans
  - extract jar in new directory, say, Import
  - create project from existing sources
  - nb does not copy these files

Importing Files

- Eclipse
  - Create new project
  - Import files directly from Jar archive

Exporting Files

- BlueJ
  - Create jar
  - Make sure to select sources
- NetBeans
  - change project properties to remove exclude for *java in Properties->Build-> Creating Jar
  - creates jar in **`dist`** subdir of project
  - Create the jar manually from **`src`** subdir of project directory

Exporting Files

- Eclipse
  - Export as Jar
  - Select the files you want to write out

Running Tests

- download junit.jar
- export to a jar (compiled classes)
- run tests on command line

```
java
     -classpath junit.jar:var-packer-assign.jar:
                                 var-packer-test.jar

VarPackerTest.AllTests
```

Running Tests

- BlueJ
  - Tools->Preferences->Libraries->add junit.jar
  - run main of AllTests class
- NetBeans
  - Import sources
  - Run Junit tests by right clicking on source file and run file, choose RunOther->Test
- Eclipse
  - Import jar
  - Run as junit

Running Tests

- If only test classes are provided
  - Run from command line