Application Programming

Week 1

Lecture 3

Deliverables

- More Java review
- Creating our first Java project in Eclipse
- Javadoc

Discussion

- Garbage collection
- This (keyword)
- Documentation

Objects

- Show the memory model for the following code

```
String fName;
fName =□Sam□;
fName = □Alex□;
String employeeName =  fName;
```

Objects

- Java will clear out old data that no variables are referencing
  - This is known as *garbage collection*
- More than one variable may refer to the same data

Self-assessment

- Which of these are primitive types, and which are the names of classes?
  - int
  - Picture
  - char
  - Double
  - Math
  - double
  - Integer
  - String

Self-assessment

- Which of these are primitive types, and which are the names of classes?
  - int
  - Picture
  - char
  - Double
  - Math
  - double
  - Integer
  - String

Methods

- There are two types of methods
  - Object methods
    - Associated with an object
    - Sent as a message to an object
    - Implicitly passed to the current object
    - Keyword: this
  - Class methods
    - Not associated with a particular object
    - Sent as a message to a class
    - Keyword: static

Self-assessment

- Which of the following lines contain an object method?

```
String greeting = "HI";
String obvious = "This is a string";
String strWithSpace = "   This is a string";
String valueOf(55);
greeting.toLowerCase();
obvious.indexOf("is");
strWithSpace.trim();
```

Self-assessment

- Which of the following lines contain an object method?

```
String greeting = "HI";
String obvious = "This is a string";
String strWithSpace = "   This is a string";
String.valueOf(55);
greeting.toLowerCase();
obvious.indexOf("is");
strWithSpace.trim();
```

Methods

- Other class methods to try
  - `Math.abs(13)`
  - `Math.abs(-13)`
  - `Math.min(3, 4)`
  - `Character.getNumericValue('A')`

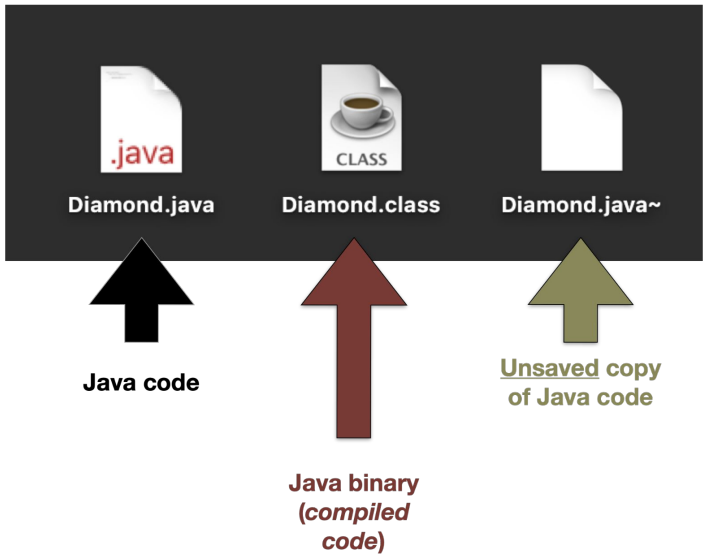
- Note there is no object associated with these methods.

Creating a project in Eclipse

- Open Eclipse
- Select/create a **workspace** (a location to save your files)
- In the **Package Explorer** (top left area of IDE), right click and select “New > Java Project”
  - Name your project (if this is for a lab, follow the lab guidelines!)
  - Ensure you’re using Java 9 (in the dropdown)
  - Uncheck the create module-info.java file
  - Click “finish”
- Create a new class by right clicking on your project name, New > Class
  - Name your class (uppercase first letter on each word)
  - To run your class, you need a **main** method. With that, you can click the green “run” button at the top of the IDE to run your program and see results in the console.

A note about files

A note about files



Javadoc Annotations

- Javadoc is a documentation generator for the Java language for generating API documentation in HTML format from Java source code.
- `@author`
  - designates the author of the code
  - belongs in the class comment
- `@param`
  - designates the parameter to a method
  - belongs in all method comments which require parameters
- `@return`
  - designates the returned value of a method
  - belongs in method comments which return values

Javadoc Annotations

```
/**
 * The Account class represents an account (…)
 *
 * @author Deitel
 * @author Deitel
 */
public class Account{
   ⋮
   /**
    * Returns the name of the account
    * @return String name of this account
    */
   public String getName(){
      return this.name;
   }
   /**
    * Sets the name of the account
    * @param n Name to set on this account (String)
    */
   public void setName(String n){
      this.name = n;
   }
}
```

Javadoc Annotation

- To generate Javadoc in Eclipse
  - Project > Generate Javadoc
  - Destination: workspace/*your_project*/doc
  - Next
  - Select all “referenced archives and projects”
  - Finish