**Java Class Generator**

For project 1 you are to create a Java class generator. Follow the directions and hints below. This is an individual project – you are NOT to work in groups; however, I fully support “helping” each other. This means that if you are asked to help someone in class then teach them, do not just supply them with code. Teaching them what you know not only helps them for the rest of their time in the course and our department but it helps you to learn.

The system, when run, will read in a text file that will act as a script. The script should contain sections used to direct your program how to create the **classes**, **interfaces**, **abstract classes**, **enumerations**, etc... For example: The below script would be used to create a Car class with properties, methods, constructors, implemented interfaces and inheritance.

Example Script file:

Note: What you can do for this system is to develop a GUI interface that can be used to

1. Create this script file
2. Read a script file and generate the classes

Also, the files that are created (3 in the example) should be structured with “sections” in the following order:

**//======…===== Properties**

**…**

**//======…===== Constructors**

**…**

**//======…===== Methods**

**…**

**//======…===== Getters / Setters**

**…**

Class:

Name: Car

Abstract: false

Extends: Rectangle

Implements: IMoveable, IDrawable

CloneMethod: true

EmptyConstructor: false

WorkhorseConstructor: true

CopyConstructor: true

Property:

Name: make

Type: String

Scope: private

Getter: true

GetterScope: public

Setter: true

SetterScope: private

Property:

Name: model

Type: String

Scope: private

Getter: true

GetterScope: public

Setter: true

SetterScope: private

Property:

Name: year

Type: String

Scope: private

Getter: true

GetterScope: public

Setter: true

SetterScope: private

End Class:

Interface:

Name: IDrawable

Method: public void draw(Graphics g);

End Interface:

Interface:

Name: IMoveable

Method: public void move();

Method: public void stop();

Method: public int speedUp();

Method: public int slowDown();

End Interface:

From a script similar to the above you are to read in each line and process it so that in the end this file will be converted into a set of classes, enumerations, and interfaces.

Below is a Utility class that can help you parse and manipulate data. Notice the keyword “static” is used. This way you can directly call upon these methods from your Tester class. An example would be if you are trying to create a setter method for a variable named “firstName”:

pw.printLn("public String get" + Utilities.camelCase("firstName") + "() {");

would save the following to the file: **public String getFirstName() {**

**import** java.util.Scanner;

**public** **abstract** **class** Utilities {

//The following methods work with the format given in the project file

//======================================================================

// A method that is give a line and returns an array of values

**public** **static** String[] parseLine(String line) {

**return** line.replace("\t","").trim().split(":");

}

// A method that reads a line from a file and returns an array of values

**public** **static** String[] parseLine(Scanner fin) {

**return** parseLine(fin.nextLine);

}

// A method that reads a line from the file and returns just the data part

**public** **static** String nextLineData(Scanner fin) {

**try** {

**return** *parseLine*(fin.nextLine())[1].trim();

} **catch**(Exception e) {

**return** "";

}

}

// A method that reads a line and returns if the value in position 1 is true

// or false. This can be used when you know you are reading a boolean line

**public** **static** **boolean** nextLineBoolean(Scanner fin) {

**return** *nextLineData*(fin).equalsIgnoreCase("true");

}

// A method that can be used to upper the first letter of a string

**public** **static** String toCamelCase(String value) {

**if**(value.length() == 0) **return** "";

**return** (value.charAt(0)+"").toUpperCase() + value.substring(1);

}

//=======================================================================

// THERE ARE MANY OTHER METHODS YOU COULD WRITE TO HELP IN YOUR CODING SO

// THIS IS THE PLACE WHERE YOU CAN CREATE THEM. HAVE FUN AND DON’T WAIT!!

//=======================================================================

}