**Assignment 1**

**Due, Thursday, June 18, 2015 for maximum 100**

**Friday, June 19, 2015 for maximum 90**

**Saturday, June 20, 2015 for maximum 80**

**Sunday, June 21, 2015 for maximum 70**

**Deliverables**

To complete this assignment you must submit your **compressed Netbeans project** to Webcourses.

**Tasks and Rubric**

|  |  |  |
| --- | --- | --- |
| Activity | | Points |
| Boggle project | Create a new Java Application project named Boggle  allowing Netbeans IDE to create the main class called Boggle |  |
| Boggle class | Create member variables of type:   1. ArrayList // stores data from data file 2. String // set to the name of the input file “BoggleData.txt” |  |
|  | Method main() should:   1. Create an instance of class ReadDataFile passing the file name variable as an argument 2. Call method populateData() from class ReadDataFile 3. Set member variable of type ArrayList equal to method call getData() from class ReadDataFile 4. Create an instance of class Board passing the file data in the ArrayList as an argument 5. Call method shakeDice() from class Board |  |
| core package | Create package **core** |  |
|  | Create class Board |  |
|  | Create class Die |  |
| Board class | Create constant fields:   1. NUMBER\_OF\_DICE equal to value 16 2. NUMBER\_OF\_SIDES equal to value 6 3. GRID equal to value 4   Create member variable of type:   1. ArrayList // stores all dice data 2. ArrayList<Die> // stores 16 dice objects   Create constructor with one parameter of type ArrayList; set the member variable of type ArrayList that stores all the Boggle data equal to the parameter in the method signature  Create method populateDice with return type void and an empty parameter list; the method should do the following:   1. Declare an variable of type class Die 2. Loop through the 16 dice:    1. Create an instance of class Die using the no-argument constructor    2. Loop through the 6 sides of the die:       1. Add each of the 6 letters to the die ArrayList representing the die letters by calling method addLetter in class Die    3. Display the letters of each die by calling method displayAllLetters() in class Die on a separate row 3. Add each die instance to the ArrayList declared specifically for class Die   Create method shakeDice with return type ArrayList and an empty parameter list; the method should do the following:   1. Call method populateDice() 2. Loop through the 16 dice, for each Die:    1. Call method getLetter in class Die    2. Display the current letter of each Die in a 4 X 4 grid    3. Return the ArrayList of Boggle dice with each letter set |  |
| Die class | Create constant field:   1. NUMBER\_OF\_SIDES equal to value 6   Create member variable of type:   1. ArrayList // stores dice data for the sides 2. String // stores the current letter of each die   Create method randomLetter with return type void and an empty parameter list; the method should do the following:   1. Declare an variable of type class Random 2. Generate a random number based on the seed value of the number of die sides 3. Set member variable representing the current letter to the data stored at the index of the random number   Create method getLetter with return type String and an empty parameter list; the method should do the following:   1. Call method randomLetter 2. Return the letter associated with the letter based on the random number   Create method addLetter with a return type of void and one parameter of type String representing one of the six letters on the die; the method should:   1. Add the passed in value to the ArrayList representing the letters on the die   Create method displayAllLetters with a return type of void and an empty parameter list; the method should:   1. Loop through all sides of the die and display the data |  |
| inputOutput package | Create package **inputOutput** |  |
|  | Create class ReadDataFile |  |
| ReadDataFile class | Define class member variables using the specified data types:   1. Scanner // for reading the file 2. String // for storing the file name 3. ArrayList // for storing the data from the file   Create constructor with one parameter of type String representing the name of the data file   1. Set local variable of type String to the value passed in   Create method populateData with return type void and an empty parameter list; it should do the following:   1. Create an instance of class URL using the file name of the data file 2. Create an instance of class File using the URL created above 3. Initialize member variable of type Scanner based on the File instance created above 4. Loop through the data file until the end    1. Add to the ArrayList representing the data in the file each value read from the data file   Create method getData with return type ArrayList and an empty parameter list that returns the ArrayList with the data from the data file |  |
| Data File BoggleData.txt | Store the downloaded data file in the inputOutput package directory on the hard drive |  |
| userInterface package | Create package **userInterface** |  |
| Boggle application |  |  |
| Test Case 1 | Test Case 1 passes |  |
| Test Case 2 | Test Case 2 passes |  |
| Test Case 3 | Test Case 3 passes |  |
|  | Source compiles with no errors |  |
|  | Source runs with no errors |  |
|  | Source includes comments |  |
| Total |  | **100** |

**Perform the following test cases**

|  |  |  |
| --- | --- | --- |
| Test Cases | | |
|  | **Action** | **Expected outcome** |
| Test Case 1 | **Project view** | Completed project view should look like figure 1 |
| Test case 2 | **Run application** | displayAllLetters should look like figure 2 |
| Test case 3 | **Run application** | shakeDice should be similar to figure 3 |

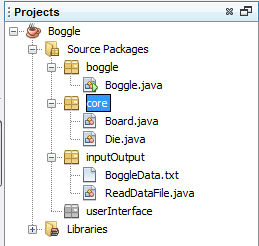
****

Figure 1 Project View

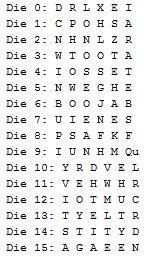


Figure 2 Output from method displayAllLetters

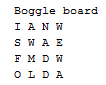


Figure 3 Output from method shakeDice