School of Computing and Mathematics
CSC-10024 - Programming I - Programming Fundamentals
Coordinator: Bappaditya Mandal
ASSESSED EXERCISE 3

ASSESSED EXERCISE 3: Return of Word Game

Deadline: To be submitted on KLE by Monday 13/December/2021, 13:00.
Weighting: 25% of the entire module mark.

For this exercise you will create a **wordGame.**

You do not need to have any prior knowledge of the game. Everything that you will need to know to complete this practical is explained below and example runs of the solution are provided at the end of this document.

Rules of the gameplay:

In this game the player is the user who enters a word at a time when prompted by the computer. The computer decides whether the player can continue or the game could be terminated. The following **five rules** must be maintained at any time of the game:

1. The **first** character of the next word should start with the **last** character of the previous word and they are case sensitive (for example, "`Eat`" and "`eat`" are different). For an invalid word, the game should continue with a message "`Entered word is invalid or does not exist in the dictionary, please try again`".
2. The word **must** **not** be entered in the game before. The game must be terminated if any word is repeated with a message "`Game over`".
3. The word should have **only** the alphanumeric characters (such as `[A-Z]` and `[a to z]`), and should **not** have any numeric character (such as numbers `4` or `7`) or space or hyphenated (such as co-worker) or special characters (such as `; , :`) within itself. For an invalid word, the game should continue with a message "`Entered word is invalid or does not exist in the dictionary, please try again`".
4. The word should be **a valid word** existing in the dictionary. Use the attached dictionary.zip (containing dictionary.txt) file as the dictionary. For an invalid word, the game should continue with a message "`Entered word is invalid or does not exist in the dictionary, please try again`".
5. If nothing is entered by the user (empty string or word with length 0), the game must **restart** from the beginning.

To achieve the rules of the above gameplay, write a `wordGame` class with:

A. Three instance variables/array:

1. `words`: An array of words listed while playing the game.
2. `outcome`: A Boolean variable which is true when game is over.
3. `continueORterminate`: A Boolean variable which is true for continuation of the message display.

B. Three instance methods

1. `gameStart`: this should check all the necessary conditions inside.
2. `wordValid`: this should check the existence of the word in the dictionary.
3. `gameRestart`: should initialize the array again and start looping again.

Additionally, write an `application` to test out the above functionalities, display all the previously entered words, prompt the user to enter a new word and play the game. This application should have some default word to start with, such as “`empathy`” and should prompt the user to enter a word for playing this game. Sensible messages should be displayed to the user as information to the user for taking appropriate action(s).

Make sure that you submit your completed project (as a zip/rar file of the entire Apache NetBeans project folder) for assessment. Ask for help if you are unsure about how to compress your project folder.

What resources might I use to get started?
Ans: Laboratory CSL3.104 and CR010, CR012 and CR014 PCs and software.
Contact: Bappaditya Mandal (Email: b.mandal@keele.ac.uk)
Office: Colin Reeves CR36, Hours: Ask during any of the practical classes in CSL3.104 or by appointment via email.

Module Learning Outcomes Assessed
1. Show practical experience of the basic concepts of computer programming.
2. Evaluate the suitability of computer language data and control structures to achieve basic problem-solving.
3. Use basic software engineering principles to design and implement computer programs.

Marking scheme (total 100%)

Description	Mark
Implementation of the class with instance variables/arrays and methods.	20%
Implementation of the five rules (each 10%, totalling to 50%) on page 1.	50%
Implementation of application with user input and display of existing words.	15%
Implementation of correct display with relevant messages.	5%
Game termination with appropriate reporting/display of messages.	10%

Example runs

Example 1:

```
Word Game started with empathy
empathy
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yellow
empathy   yellow
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: wipes
empathy   yellow    wipes
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: shy
empathy   yellow    wipes     shy
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yellow
Game over
```

In the above example, the game was terminated because the user entered a word which was already existing in the list (such as '`yellow`').

Example 2:

```
Word Game started with empathy
empathy
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yes
empathy   yes
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...:
Game restarting as the entered word was empty...Word game
restarted with empathy
empathy
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...:
```

In the above example, the game restarted in between because the user entered an empty word (such as nothing was entered when prompted).

Example 3:

```
Word Game started with empathy
empathy
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yess
Entered word is invalid or does not exist in the
dictionary, please try again
empathy
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yes
```

```
empathy   yes
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: say
empathy   yes  say
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yes
Game over
```

In the above example, the word (for example, '`yess`') does not exist in the dictionary, so it prompted the user to try again.

Example 4:

```
Word Game started with empathy
empathy
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yes
empathy   yes
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: sai d
Entered word is invalid or does not exist in the
dictionary, please try again
empathy   yes
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: s:,ed
Entered word is invalid or does not exist in the
dictionary, please try again
empathy   yes
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: se:ed
Entered word is invalid or does not exist in the
dictionary, please try again
empathy   yes
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: seed-ed
Entered word is invalid or does not exist in the
dictionary, please try again
empathy   yes
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: say
empathy   yes  say
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yellow:
Entered word is invalid or does not exist in the
dictionary, please try again
empathy   yes  say
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yellow
empathy   yes  say  yellow
```

```
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: wet
empathy   yes  say  yellow    wet
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: toss
empathy   yes  say  yellow    wet  toss
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: say
Game over
```

In the above example, the user is informed that word is invalid and prompted to enter another word. Several cases are tried such as ‘`sai d`’, ‘`s:,ed`’, ‘`se:ed`’ and ‘`yellow:`’, etc. Kindly note that the examples shown here are not exhaustive and do not cover all the cases. The game is terminated when the word ‘`say`’ is repeated.

Example 5:

```
Word Game started with empathy
empathy
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: Yes
Entered word is invalid or does not exist in the
dictionary, please try again
empathy
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yes
empathy   yes
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: say
empathy   yes  say
Enter a word (caseSensitive) to play the game, such as
empathy->yet->top...: yes
Game over
```

In the above example, ‘`empathy`’ ended with ‘`y`’, so ‘`Yes`’ is an invalid word (case sensitive), whereas ‘`yes`’ is valid.

Kindly note that the examples shown above are not exhaustive and do not cover all the cases.