PECANWOOD

COLLEGE

Prepared for Life

INFORMATION TECHNOLOGY JULY PRACTICAL EXAMINATION GRADE 11

NAME: _____ **GRADE:** _____

DATE: 12 JULY 2022
MODERATOR: MR C SEEWALD
EXAMINER: MR SC EILERTSEN

MARKS: 110
TIME: 180 MINUTES

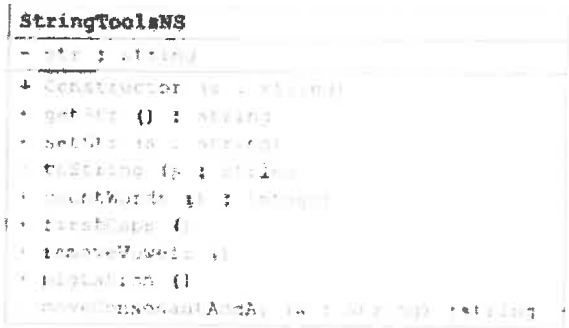
INSTRUCTIONS:

1. This examination paper is made up of 7 pages.
2. Note the mark allocation for each question.
3. The screen shots in each question are part of the question.
4. You will be provided with two text files "fourExitsB.txt: and "nationality.txt"
5. There is some additional paper at the end of the examination for any notes you may want to make as you code your solution.
6. Marks are awarded for indentation, variable names, class name, good use of white space, comments, and good readable layout (4 marks)
7. Save your work on a regular basis.
8. Question one. This assignment has already been handed in.
9. For question two you may use the two pages of notes that you have prepared from Learning Unit 6 in your textbook (pages 98 to 107 only). Only notes pertaining to the chapter "Date and Time" may be written on these pages. Your notes must be handed in for moderation the day before the exam. **Only notes signed, and with the school stamp are approved for use with this examination.**
10. Beside question two you may use the help files and language prompts that are part of the IDE you are using (NetBeans for example) but you may not use any other help. You are not allowed to access the internet.

Section One **Creating your own reusable class**

This question is a pre-examination assignment. Complete the task and submit your work online using Ms Teams on the due date 17 June 2022.

You have been provided with "Learning Unit 7 Creating Reusable Classes" - Exploring IT Java Grade 11. You must work through this chapter from page 108 to page 119 and code Activities 1 to 5 as showcased in the text as well as Exercise 2 question 3 (3.1, 3.2, 3.3).

Create a new project called LU7. Create a new package called lu7 Create two classes - A method class and a UI class The **method class** must be called StringToolsNS and must use the class diagram alongside.	

Each method in the method class must only do one task so that your solution matches the class diagram above. (10)

Here is a summary of the tasks outline in the textbook.

- Count the names in the full name.
- Make the first letter of every name into a capital letter.
- Print the full name having removed all the vowels.
- Print a Pig Latin version of the full name.
- Print out the full names, name by name, with each name being numbered from one to n.
- Add Exercise 2 question 3 (3.1, 3.2, 3.3) to your StringToolsNS class - determine if the name has any invalid characters

Here is an example of input and output

INPUT FROM KEYBOARD:

steve carl johan bergin alphonso eilertsen

OUTPUT:
The full name is: steve carl johan bergin alphonso eilertsen
Name count is 6
The new name is: Steve Carl Johan Bergin Alphonso Eilertsen
The name without vowels is: Stv Crl Jhn Brgn Alphns Elrtsn
The pig latin version is: tevesay arlcay ohanjay erginbay alphonsoay eilertsenay
Name 1 is steve
Name 2 is carl
Name 3 is johan
Name 4 is bergin
Name 5 is alphonso
Name 6 is eilertsen
Name is valid - no illegal characters were found

A **UI class** with a main method ("StringToolsUI" - not "TestStringToolsNS" as per the text book)

The main method must accept a full long name as input from the keyboard. In each case the main method will pass the full name as a parameter to the relevant method in the method class. The method class must have a constructor that accepts the full name from the main class. (10)

[20]

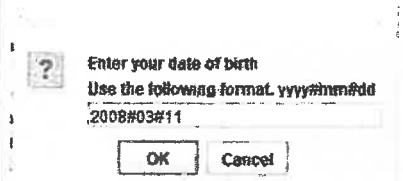
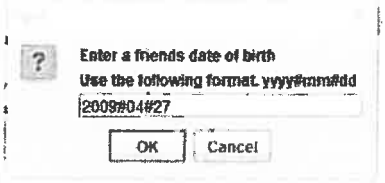
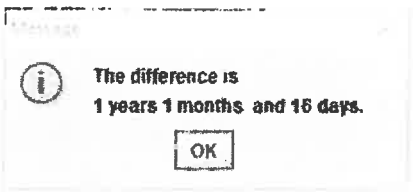
Section Two **Date and time**

For this question you may use the two pages of notes that you have prepared from Learning Unit 6 in your textbook (pages 98 to 107 only). Only notes pertained to the chapter "Date and Time" may be written on these notes. Your notes must be handed in for moderation the day before the exam. **Only notes signed, and with the school stamp, are approved for use with this examination.**

Write a **one class program** with static variables and four static methods to calculate the age difference between you and a friend.

Your methods should be as follows . . .

- The first static method must get your date of birth from the keyboard.
- The second static method must get your friend's date of birth from the keyboard.
- The third static method must report the difference in years, months, and days using a useful output message.
- The main method that calls the other methods.

Method One	Method Two	Method Three
		

(14)

Section Three **Reading from a text file – no delimiters using the Scanner class**

At a busy intersection with many vehicles per day, there are four exits – exit 1, exit 2, exit 3 and exit 4

A video camera linked to a single board computer is mounted on a high pole. The camera makes a note of which exit was used by each and every vehicle; this number of the exit used is written to a text file.

At the end of the day the text file is uploaded to the traffic control centre and the program is reset back to zero.

Closely examine the text file "fourExitsB.txt" as provided. This is the text file from a single 24-hour period.

Write a **one class program** with static variables and static methods to read in the text file. Your program must count how many vehicles used exit 1, how many used exit 2, how many used exit 3, and how many used exit 4 (again have a look at the text file).

During your bare bones approach to this question, write a method that will print out the whole array. That way you will know that your array is working correctly.

Write methods so that your output looks like the screen shot below.

```
Four exits
==========
Exit 1:   277
Exit 2:   286
Exit 3:   304
Exit 4:   284
==========
Biggest exit is Exit 3 with a value of 304
```

Your methods should be as follows . . .

- One method to read in the whole text file into a suitable array.
- One method to print out the whole array (**output not shown here**).
- One method to report the values for each exit.
- One method to determine the exit with the highest number of vehicles.
- Main method that calls the methods in the right order.

(32)

Section Four **Reading from a text file – with delimiters using the Scanner class**

Here is a text file of South African citizens and Botswana citizens – "nationality.txt" as provided.

The fields for South Africa citizens are as follows . . .

- Passport number. Name. Tax payer. Currency

The fields for Botswana citizens are as follows . . .

- Passport number. Name.. Currency

```
FGD-ZA-7674884A#andrew alpha#T#76567.76
6672889-BO#erric echo#654567.71
876553-BO#cindy charlie#465.67
GFD-ZA-9565776C#betty bravo#F#75769.37
3456278-BO#foxy foxtrot#6465476.87
6672889-BO#david dandy#6750.65
9876567-BO#gary golf#65576.67
NBV-ZA-9763537B#inky india#T#7869.76
 DR-ZA-4256297A#jacky jackson#F#86876.01
NBV-ZA-6697899A#henry hotel#T#87879.78
```

The South Africans have a ZA in their passport numbers while those from Botswana have a BO.

Create a **one class program** with static variables and methods to read in the text file and output the result to the monitor. By using the String method indexOf("-") and substring, you will be able to copy all the South Africans into a South African array and all the Botswana citizens into a Botswana array.

Your methods should be as follows . . .

- One method to read in the text file and create one array for South Africans and one for Botswana citizens.
- One method to display the South African citizens.
- One method to display the Botswana citizens.
- One method to display the total amount of currency from both South Africans and Botswana citizens together. It is a good idea to try to get this total amount in the while loop that reads in the text file, line by line. This method should only need to display the amount, but this is just a suggestion.
- Main method that calls the methods in the right order.

Here is the expected output – South African passport holders followed by Botswana passport holders

```
Country ZA
=========================================================
Passport                  Name            Tax       Amount
=========================================================
FGD-ZA-7674884A           ANDREW ALPHA    true      76567.76
GFD-ZA-9565776C           BETTY BRAVO     false     75769.37
NBV-ZA-9763537B           INKY INDIA      true      7869.76
DDR-ZA-4256297A           JACKY JACKSON   false     86876.01
NBV-ZA-6697899A           HENRY HOTEL     true      87879.78
Country BO
=========================================
Passport       Name            Amount
=========================================
6672889-BO     ERRIC ECHO      654567.71
876553-BO      CINDY CHARLIE   465.67
3456278-BO     FOXY FOXTROT    6465476.87
6672889-BO     DAVID DANDY     6750.65
9876567-BO     GARY GOLF       65576.67

=========================================
Total Amount - ZA and BO - R7527800.25                          (40)
```

TOTAL: 110 marks

Additional Paper for notes you may want to make as you code.

Additional Paper for notes you may want to make as you code.

```java
// Learning unit 7. Creating reuseable classes
// String and Character handling
// The method class
// Activity 4 and 5 and individual words into array

public class StringToolsNS6 {

   // Global variable
   private String fullName;

   public StringToolsNS6(String fn) {
   // If a value arrives via the constructor, and is saved into the global variable,
   // then all the methods have access to it.
      fullName = fn;

   } // end constructor ==========================================

   public String getName() {

      return fullName;
   }// end getStr ================================================

   public void setName (String fn) {
      fullName = fn;

   } // end setStr ================================================

   public String toString () {


      return "The full name is: " + fullName;

   } // end toString ==============================================

   public int countNames() {
   // A word - starts with a letter, while the character before is not a letter

      int count = 0;
      char previous, current;
      String temp = " " + fullName;

      // Need to start with the second character, and then check the character before it.
      for(int i = 1; i < temp.length(); i++) {
         current = temp.charAt(i);
         previous = temp.charAt(i-1);

         // isLetter is a class method - a static Character method
         if(Character.isLetter(previous) == false && Character.isLetter(current) == true)
            count = count + 1;

      } // end for

      return count;
   } // end countWords ===========================================

   public String firstCaps() {
   // First letter of every word must be PERMANENTLY changed to a capital letter.
   // Change the whole sentence to lowercase.
   // Build a new sentence character by character

      String temp = " " + fullName;
      temp = temp.toLowerCase();
      String newStr = "";
      char previous, current;

      // Need to start with the second character, and then check the character before it.
      for(int i = 1; i < temp.length(); i++) {
```

```
68          current = temp.charAt(i);
69          previous = temp.charAt(i-1);
70
71          // isLetter is a class method - a static Character method
72          if(Character.isLetter(previous) == false && Character.isLetter(current) == true)
73             newStr = newStr + Character.toUpperCase(current);
74          else
75             newStr = newStr + current;
76
77          // The global variable is changed permanently
78          fullName = newStr;
79
80       } // end for loop
81
82       return fullName;
83
84    } // countCaps ============================================
85
86    public String removeVowels() {
87    // Remove vowels unless it is the first letter of the word
88
89       String temp = " " + fullName;
90       String newStr = "";
91       char previous, current;
92       final String VOWELS = "aeiouAEIOU";
93
94       for(int i = 1; i<=temp.length() - 1; i ++) {
95          previous = temp.charAt(i - 1);
96          current = temp.charAt(i);
97
98          // AND is evaluated before the OR.
99          // Add to string if not a vowel or beginning of a word
100         // indexOf returns -1 is the argument is not found
101         if(Character.isLetter(previous) == false &&
102            VOWELS.indexOf(current) >= 0 ||
103            VOWELS.indexOf(current) == - 1)
104               newStr = newStr + current;
105
106      } // end for
107
108      return newStr;
109   } // end removeVowels ======================================
110
111   public String pigLatin () {
112
113      String singleWord;
114      // Need a space before and after to create previous and current positions
115      String temp = " " + fullName + " ";
116      temp = temp.toLowerCase();
117      String pigStr = " ";
118      char previous, current;
119      int wordBegin = 0;
120
121      for(int i = 1; i < temp.length(); i++)  {
122         previous = temp.charAt(i-1);
123         current = temp.charAt(i);
124
125         // find the beginning position and end position of each word
126         if(Character.isLetter(previous) == false &&
127            Character.isLetter(current) == true)
128               wordBegin = i;
129         if(Character.isLetter(previous) == true &&
130            Character.isLetter(current) == false) {
131               singleWord = temp.substring(wordBegin, i);
132               pigStr = pigStr + moveConsonantAddAy(singleWord) + " ";
133
134            } // end if
```

```
135
136       } // end for
137
138       return pigStr;
139
140    } // end pigLation ========================================
141
142      public void eachWord () {
143
144       String singleWord;
145       // Need a space before and after to create previous and current positions
146       String temp = " " + fullName + " ";
147       temp = temp.toLowerCase();
148       String pigStr = " ";
149       char previous, current;
150       int wordBegin = 0;
151       String[] wordArr = new String[50];
152       int numberOfWord = 0;
153
154       for(int i = 1; i < temp.length(); i++)  {
155          previous = temp.charAt(i-1);
156          current = temp.charAt(i);
157
158          // find the beginning position and end position of each word
159          if(Character.isLetter(previous) == false &&
160             Character.isLetter(current) == true)
161                wordBegin = i;
162          if(Character.isLetter(previous) == true &&
163             Character.isLetter(current) == false) {
164                singleWord = temp.substring(wordBegin, i);
165                wordArr[numberOfWord] = singleWord;
166                numberOfWord++;
167
168             } // end if
169
170       } // end for
171
172       for(int i = 0; i < numberOfWord; i++)
173             System.out.println("Name " + (i+1) + " is " + wordArr[i]);
174
175    } // end pigLation ========================================
176
177
178    // Private helper method for method pigLatin
179    private String moveConsonantAddAy (String w) {
180
181       final String VOWELS = "aeiouAEIOU";
182
183       // if the first letter is not a vowle
184       if(VOWELS.indexOf (w.charAt(0)) == - 1)
185       // move the consonant to the end of the word
186          w = w.substring(1) + w.charAt(0);
187
188       w = w + "ay";
189
190       return w;
191
192    } // end moveConsonantAddAy ================================
193
194 } // end class
```

NB Finding if an element is present in a list.

10

```
1 // Learning unit 7. Creating reuseable classes
2 // String and Character handling
3 // The UI class
4 // Activity 4 and 5 and individual words into array
5
6 import java.util.Scanner;
7
8 public class TestStringToolNS6 {
9
10    public static void main(String[]args) {
11
12       String theString = null, noVowels = null;
13       int wordCount = 0;
14       Scanner sentence = new Scanner(System.in);
15       System.out.println("Enter the full name");
16       theString = sentence.nextLine();
17       StringToolsNS6 stns = new StringToolsNS6(theString);
18
19       theString = stns.toString();
20
21       System.out.println(theString);
22       wordCount = stns.countNames();
23       System.out.println("Word count is " + wordCount);
24
25       String newString = stns.firstCaps();
26       System.out.println("The new string is: " + newString);
27
28       noVowels = stns.removeVowels();
29       System.out.println("The sentence without vowels is: " + noVowels);
30
31       String stringPig = stns.pigLatin();
32       System.out.println("The pig latin version is: " + stringPig);
33
34       stns.eachWord();
35    } // end main
36 }
```

object + parameter

Ques 2.

```
/*
 * Difference in ages
 */
package julyexampractical;

import java.time.*;
import javax.swing.JOptionPane;

public class MyAgeToday {

    private static LocalDate myDob;
    private static LocalDate FriendDob;
    private static Period diff;
    private static int year = 0, month = 0, day = 0;

    public static void main(String[]args){

        getYourDateOfBirth();
        getFriendsDateOfBirth();
        calculateDifference();

    } // end main

    private static void getYourDateOfBirth(){
        String myDateOfBirth = JOptionPane.showInputDialog(null, "Enter your
date of birth" + "\n" +
                "Use the following format. yyyy#mm#dd");
        year = Integer.parseInt(myDateOfBirth.substring(0,4));
        month = Integer.parseInt(myDateOfBirth.substring(5,7));
        day = Integer.parseInt(myDateOfBirth.substring(8));
        myDob = LocalDate.of(year, month, day);

    }

    private static void getFriendsDateOfBirth() {
        String FriendDateOfBirth = JOptionPane.showInputDialog(null, "Enter a
friends date of birth" + "\n" +
                "Use the following format. yyyy#mm#dd");
        year = Integer.parseInt(FriendDateOfBirth.substring(0,4));
        month = Integer.parseInt(FriendDateOfBirth.substring(5,7));
        day = Integer.parseInt(FriendDateOfBirth.substring(8));
        FriendDob = LocalDate.of(year, month, day);

    }

    private static void calculateDifference() {

        diff = Period.between(myDob, FriendDob);
        JOptionPane.showMessageDialog(null, "The difference is " + "\n" +
diff.getYears() + " years " + diff.getMonths() + " months " +
            " and " + diff.getDays() + " days.");

    }
}
```

global variables

match methods

string to int

x | 14

Ques 3

```
/*
 * Four Exits
 */
package julyexampractical;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

public class FourExits {

    private static int[] exits = new int[1500];
    private static int counter;
    private static int exit1 = 0,exit2 = 0, exit3 = 0, exit4 = 0;

    public static void main(String[]args) {
        readFile();
        // displayAll();
        System.out.println("Four exits");
        System.out.println("=============");
        reportExits();
        biggestExit();
    }

    private static void readFile() {

        int exitInt = 0;
        try {
            Scanner scFile = new Scanner(new File("fourExitsB.txt"));
            counter = 0;

            while(scFile.hasNext()){
                String line = scFile.next();
                exitInt = Integer.parseInt(line);
                if(exitInt == 1)
                    exit1++;
                else if(exitInt == 2)
                    exit2++;
                else if (exitInt == 3)
                    exit3++;
                else if (exitInt == 4)
                    exit4++;
                else {
                    System.out.println("error with data in file");
                    System.exit(0);
                }

                exits[counter] = exitInt;
                counter++;
            } // end while

        } catch (FileNotFoundException ex) {
```

```
            Logger.getLogger(FourExits.class.getName()).log(Level.SEVERE, null,
ex);
            System.out.println("File not found");
        }

    } // end readFile

    private static void displayAll() {

        System.out.println("Display all");
        for(int i = 0; i < counter; i++) {

            System.out.println(exits[i]);
        } // end for

    } // end displayAll

    private static void reportExits(){

        System.out.println("Exit 1:" + "\t" + exit1);
        System.out.println("Exit 2:" + "\t" + exit2);
        System.out.println("Exit 3:" + "\t" + exit3);
        System.out.println("Exit 4:" + "\t" + exit4);
        System.out.println("==============");
    } // end report Exits

    private static void biggestExit() {

        int biggest = exit1;
        String theExit = "Exit 1 ";

        if(exit2 > biggest){
            biggest = exit2;
            theExit = "Exit 2 ";
        }
        if(exit3 > biggest) {
            biggest = exit3;
            theExit = "Exit 3 ";

        }
        if(exit4 > biggest){
            biggest = exit4;
            theExit = "Exit 4 ";

        }

        System.out.println("Biggest exit is " + theExit + " with a value of " +
biggest);

    } // end biggestExit
}
```

x | 32

Ques 4

```
/*
 * Nationality
 */
package julyexampractical;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Nationality {

    private static String[] countryZA = new String[50];
    private static String[] countryBO = new String[50];
    private static int counterZA = 0, counterBO = 0;
    private static double totalAmount = 0.0;

    public static void main(String[]args){

        readFile();
        displayZA();
        displayBO();
        displayAmount();
    } // end main

    private static void readFile(){

        String passport = null, country = null, name = null;
        boolean tax = false;
        double amount = 0.0;

        try {
            Scanner scFile = new Scanner(new File("nationality.txt"));

            while(scFile.hasNext()) {
                String line = scFile.nextLine();
                Scanner scLine = new Scanner(line).useDelimiter("#");
                passport = scLine.next();
                int dash = passport.indexOf("-");
                country = passport.substring(dash + 1, dash + 3);

                if(country.equals("ZA")){
                    name = scLine.next();
                    name = name.toUpperCase();
                    tax = scLine.nextBoolean();
                    amount = scLine.nextDouble();
                    totalAmount = totalAmount + amount;
                    countryZA[counterZA] = passport + "\t" + "\t" + name + "\t"
+ tax + "\t" + amount;
                    counterZA++;
                }

                if(country.equals("BO")) {
```

```
                    name = scLine.next();
                    name = name.toUpperCase();
                    amount = scLine.nextDouble();
                    totalAmount = totalAmount + amount;
                    countryBO[counterBO] = passport + "\t" + name + "\t" +
amount;

                    counterBO++;
                }

            } // end while

        } catch (FileNotFoundException ex) {
            Logger.getLogger(Nationality.class.getName()).log(Level.SEVERE,
null, ex);
            System.out.println("Error with file");
        }
    } // end readFile

        private static void displayZA() {


System.out.println("=============================================================");
            System.out.println("Country ZA");

System.out.println("=============================================================");
            for(int i = 0; i < counterZA;i++) {

                System.out.println(countryZA[i]);
            } // end for

            System.out.println();
        } // end displayZA

        private static void displayBO() {

            System.out.println("=============================================");
            System.out.println("Country BO");
            System.out.println("=============================================");
            for(int i = 0; i < counterBO; i++) {
                System.out.println(countryBO[i]);
            }

            System.out.println();
        } // end displayBO

        private static void displayAmount() {

            System.out.println("=============================================");
            System.out.println("Total Amount - ZA and BO - R" + totalAmount);

        } // end displayAmount

} // end class
```

No boolean

x / 40