Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації

і управління

Звіт

з лабораторної роботи № 2 з дисципліни

«Програмування - 2. Структури даних та алгоритми»

«ДОСЛІДЖЕННЯ ОПИСУ ТА ВИКОРИСТАННЯ КЛАСІВ»

Виконав студент: ІС-03 Філоненко Андрій Ігорович

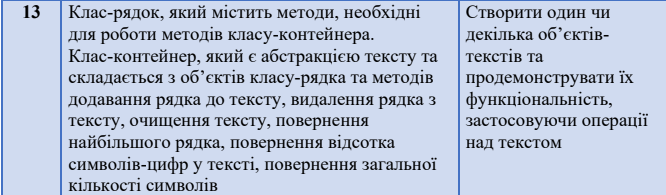
(шифр, прізвище, ім'я, по батькові)

Перевірив викладач: Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2021

1. **Умова задачі**



1. **Текст файлів проекту**

**C++ main:**

#include<iostream>  
#include"myClass.h"  
#include<iomanip>  
  
using namespace std;  
  
  
void show(Text & obj) {  
 for (int k = 0; k < 4; k++){  
 for (int i = 0; i < obj.rows[k].rSize; i++)  
 cout << obj.rows[k].data[i];  
 cout << endl;  
 }  
}  
  
  
int main(){  
 Text myStr(5);  
 myStr.addRow((char \*) "Hello");  
 myStr.addRow((char \*) "Next is digits");  
 myStr.addRow((char \*) "123456789");  
 myStr.addRow((char \*) "Goodbye");  
 cout << endl << "========STARTING TEXT========" << endl << endl;  
 show(myStr);  
 cout << endl << endl;  
  
 cout << endl << "========CALCULATIONS========" << endl << endl;  
 cout << "Amount of characters in the text : " << myStr.getCharAmount() << endl;  
 cout << "Digit Percentage : " << setprecision(3) << myStr.getDigPercentage() << "% | " << myStr.getDigAmount();  
 cout << " digits to " << myStr.getCharAmount() << " characters " << endl;  
 cout << "Max Row : ";  
 for ( int i = 0 ; i < myStr.rows[myStr.getMaxRow()].rSize; i++){  
 cout << myStr.rows[myStr.getMaxRow()].data[i];  
 }  
 cout << endl;  
  
 cout << endl << "========ERASING TEXT========" << endl << endl;  
 cout << "Text after deleting row 1 : " << endl;  
 myStr.delRow(1);  
 show(myStr);  
 myStr.erRows();  
 cout << "Text after erasing it completely : " << endl;  
 show(myStr);  
 cout << endl << "========END========" << endl << endl;  
};

**C++ cpp:**

#include <iostream>  
#include "myClass.h"  
using namespace std;  
  
void RowContainer::fill\_row(char \* inp) {  
 rSize = strlen(inp);  
 data = new char [rSize];  
 for (int i = 0; i < strlen(inp); i++)  
 data[i] = inp[i];  
}  
  
Text::Text(int size) {  
 rows = new RowContainer [size];  
}  
  
void Text::addRow(char \* inp) {  
 rows[filled\_rows].fill\_row(inp);  
 filled\_rows++;  
}  
  
  
void Text::delRow(int index) {  
 for (int i = index; i < filled\_rows; i++) {  
 rows[i] = rows[i+1];  
 }  
 rows[filled\_rows].data = nullptr;  
 rows[filled\_rows].rSize = 0;  
 filled\_rows--;  
}  
  
void Text::erRows() {  
 for (int i = 0; i < filled\_rows; i++) {  
 rows[i].data = {};  
 rows[i].rSize = 0;  
 }  
 filled\_rows = 0;  
}  
  
int Text::getCharAmount(){  
 int charCounter{};  
 for (int i = 0; i < filled\_rows; i++){  
 charCounter+=rows[i].rSize;  
 }  
 return charCounter;  
}  
  
int Text::getDigAmount() {  
 double cnt{};  
 for (int i = 0; i < filled\_rows; i++){  
 for (int j = 0 ; j < rows[i].rSize; j++){  
 if(isdigit(rows[i].data[j])){  
 cnt++;  
 }  
 }  
 }  
 return cnt;  
}  
double Text::getDigPercentage(){  
 double percentage = (double)getDigAmount()/(double)getCharAmount()\*100;  
 return percentage;  
}  
  
int Text::getMaxRow(){  
 int index{};  
 for (int i = 0; i < filled\_rows; i++){  
 if (rows[i].rSize > rows[index].rSize)  
 index = i;  
 }  
 return index;  
}

**C++ header:**

#pragma once  
  
class RowContainer {  
public:  
 void fill\_row(char \* inp);  
  
 ~RowContainer() {  
 delete[] data;  
  
 }  
  
 char \* data {};  
 int rSize {};  
};  
  
class Text {  
public:  
 explicit Text(int size);  
 ~Text(){  
 delete[] rows;  
 }  
  
 void addRow(char \* inp);  
  
 void delRow(int ind);  
 void erRows();  
 int getCharAmount();  
 int getDigAmount();  
 double getDigPercentage();  
 int getMaxRow();  
 int filled\_rows {};  
 RowContainer \* rows {};  
};

**C# main:**

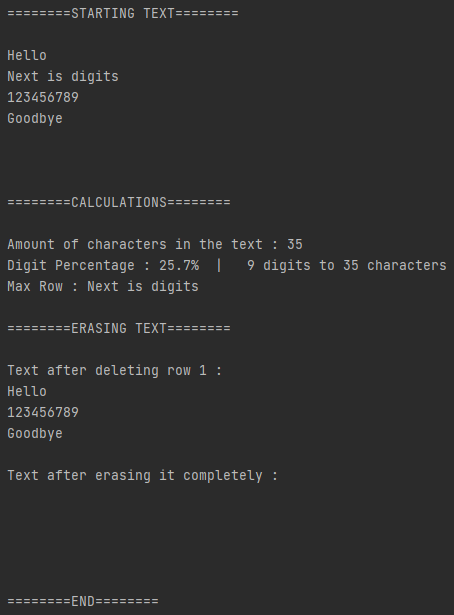
using System;  
using myStringLib;  
  
namespace lab2  
{  
 internal static class Program  
 {  
 private static void Render(ref Text obj)  
 {  
 for (var k = 0; k < 3; k++){  
 for (var i = 0; i < obj.Rows[k].rSize; i++)  
 Console.Write(obj.Rows[k].data[i]);  
 Console.WriteLine();  
 }  
 }  
   
 internal static void Main(string[] args)  
 {  
 var myStr = new Text(5);  
 for (var i = 0; i < 5; i++)  
 {  
 myStr.Rows[i] = new RowContainer();  
 }  
 myStr.addRow("Hello".ToCharArray());  
 myStr.addRow("Next is digits".ToCharArray());  
 myStr.addRow("123456789".ToCharArray());  
 myStr.addRow("Goodbye".ToCharArray());  
 Console.WriteLine("========STARTING TEXT========");  
 Render(ref myStr);  
 Console.WriteLine();  
   
 Console.WriteLine("========CALCULATIONS========");  
 Console.WriteLine("Amount of characters in the text : " + myStr.getCharAmount());  
 Console.WriteLine("Amount of digits in the text : " + myStr.getDigAmount());  
 Console.WriteLine("Percentage of digits : " + Math.Round(myStr.getDigPercentage(), 3) + "%");  
 Console.Write("Max Row : ");  
 for ( int i = 0 ; i < myStr.Rows[myStr.getMaxRow()].rSize; i++)  
 {  
 Console.Write(myStr.Rows[myStr.getMaxRow()].data[i]);  
 }  
 Console.WriteLine();  
   
 Console.WriteLine("========ERASING TEXT========");  
 Console.WriteLine("Row 1 deleted : ");  
 myStr.delRow(1);  
 Render(ref myStr);  
 Console.WriteLine();  
 Console.WriteLine("Everything deleted : ");  
 myStr.erRows();  
 Render(ref myStr);  
 Console.WriteLine();  
   
 }  
 }  
 }

**C# lib:**

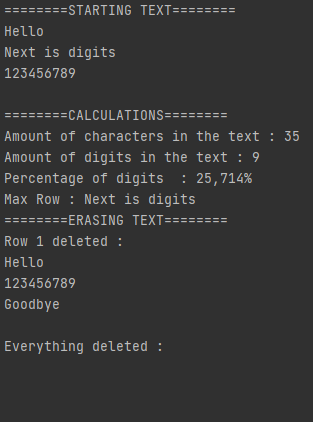
using System;  
  
namespace myStringLib  
{  
 public class RowContainer  
 {  
 public void fill\_row(char [] inp)  
 {  
 rSize = inp.Length;  
 data = new char [rSize];  
 for (var i = 0; i < inp.Length; i++)  
 data[i] = inp[i];  
 }  
  
 public char [] data;  
 public int rSize;  
 }  
  
 public class Text  
 {  
 private int \_filledRows;  
 public readonly RowContainer[] Rows;  
  
 public Text(int size)  
 {  
 Rows = new RowContainer [size];  
 }  
  
 public void addRow(char[] inp)  
 {  
 Rows[\_filledRows].fill\_row(inp);  
 \_filledRows++;  
 }  
  
 public void delRow(int index)  
 {  
 for (var i = index; i < \_filledRows; i++)  
 {  
 (Rows[i], Rows[i + 1]) = (Rows[i + 1], Rows[i]);  
 }  
  
 Rows[\_filledRows].data = "".ToCharArray();  
 Rows[\_filledRows].rSize = 0;  
 \_filledRows--;  
 }  
  
   
  
 public void erRows()  
 {  
 for (var i = 0; i < \_filledRows; i++)  
 {  
 Rows[i].data = "".ToCharArray();  
 Rows[i].rSize = 0;  
 }  
  
 \_filledRows = 0;  
 }  
  
 public int getCharAmount()  
 {  
 int charCounter = 0;  
 for (int i = 0; i < \_filledRows; i++){  
 charCounter+=Rows[i].rSize;  
 }  
 return charCounter;  
 }  
  
 public double getDigAmount()  
 {  
 double cnt = 0;  
 for (int i = 0; i < \_filledRows; i++){  
 for (int j = 0 ; j < Rows[i].rSize; j++){  
 if(Char.IsDigit((Rows[i].data[j]))){  
 cnt++;  
 }  
 }  
 }  
 return cnt;  
 }  
  
 public double getDigPercentage()  
 {  
 return (getDigAmount() / (double) (getCharAmount()) \* 100);  
 }  
  
 public int getMaxRow()  
 {  
 int index = 0;  
 for (int i = 0; i < \_filledRows; i++){  
 if (Rows[i].rSize > Rows[index].rSize)  
 index = i;  
 }  
 return index;  
 }  
  
  
 }  
}

1. **Екранні форми результату**

**С++**



**C#:**



1. **Висновки**

Дана програма працює коректно та виконує поставлену задачу. Реалізовано на С++ та С#. Створено два класи-абстракції : тексту та рядку.

Реалізовано певні методи, що наведено у завданні.