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 №4 з дисципліни

«Програмування - 2. Структури даних та алгоритми»

«ДОСЛІДЖЕННЯ ПЕРЕВАНТАЖЕННЯ ОПЕРАЦІЙ ТА

ОПЕРАТОРІВ»

Виконав студент: ІС-03 Філоненко Андрій Ігорович

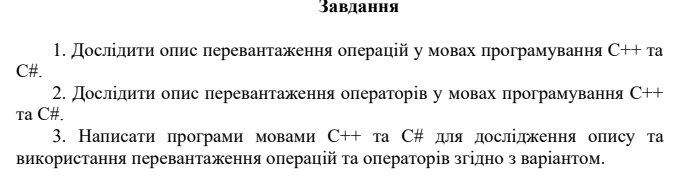
(шифр, прізвище, ім'я, по батькові)

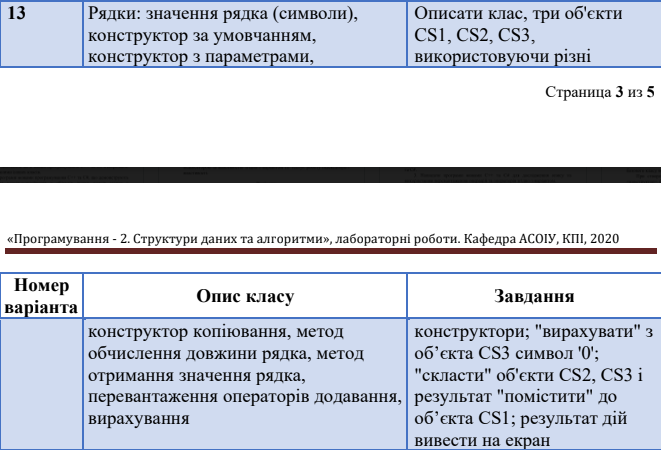
Перевірив викладач: Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2021

1. **Умова задачі**

****



1. **Текст файлів проекту**

Program.cs

using System;  
  
namespace cString\_lab4  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 cString object1 = new cString();  
 cString object2 = new cString("Hello");  
 cString object3 = new cString("By00e");  
 Console.WriteLine("==================");  
   
 //  
  
 object1.printString();  
 object2.printString();  
 object3.printString();  
   
 // +  
   
 Console.WriteLine("==================");  
 Console.WriteLine("Object 1 after addition of object 2 and object 3 : ");  
 object1 = object2 + object3;  
 object1.printString();  
   
 // -  
   
   
 Console.WriteLine("==================");  
 Console.WriteLine("Object 3 after erasing character : ");  
 object3 -= '0';  
 object3.printString();  
 Console.WriteLine("==================");  
 }  
 }  
}

cStringClass.cs

using System;  
using System.Linq;  
  
namespace cString\_lab4  
{  
 internal class cString  
 {  
 private char[] data;  
  
 public cString()  
 {  
 data = null;  
 }  
  
 public cString(string text)  
 {  
 data = text.ToCharArray();  
 }  
  
 public cString(cString obj)  
 {  
 this.data = obj.data;  
 }  
  
 public int strSize()  
 {  
 return data.GetLength(0);  
 }  
  
 char[] getData()  
 {  
 return data;  
 }  
  
 public static cString operator +(cString str1, cString str2)  
 {  
 cString buffString = new cString();  
 buffString.data = new char[str1.data.GetLength(0) + str2.data.GetLength(0)];  
 int indexer = 0;  
 for (int i = 0; i < str1.strSize(); i++)  
 {  
 buffString.data[indexer] = str1.data[i];  
 indexer++;  
 }  
 for (int i = 0; i < str2.strSize(); i++)  
 {  
 buffString.data[indexer] = str2.data[i];  
 indexer++;  
 }  
  
 return buffString;  
 }  
  
 public static cString operator -(cString str, char symb)  
 {  
 cString buffString = new cString();  
 buffString.data = new char[str.data.GetLength(0) - 1];  
 int indexer = 0;  
 if (str.data.Contains(symb))  
 {  
 for (int i = 0; i < str.strSize(); i++)  
 {  
 if (symb == str.data[i])  
 {  
 continue;  
 }  
  
 buffString.data[indexer] = str.data[i];  
 indexer++;  
 }  
  
 return buffString;  
 }  
  
 throw new Exception($"There's no {symb} characters in this string");  
 }  
  
 public void printString()  
 {  
 Console.WriteLine(data);  
 }  
 }  
}

main.cpp

#include <iostream>  
#include"cString.h"  
  
int main(){  
 cString object1 = cString();  
 cString object2 = cString((char\*)"Hell0");  
 cout << "==================" << endl;  
  
 //  
  
 object2.printString();  
  
 cString object3 = cString(object2);  
 object3.printString();  
  
  
 object2 = object2 - '0';  
 object2.printString();  
  
 cout << "==================" << endl;  
 cout << "Object 1 after addition of object 2 and object 3 : " << endl;  
 object1 = object2 + object3;  
 object1.printString();  
  
  
}

cString.h

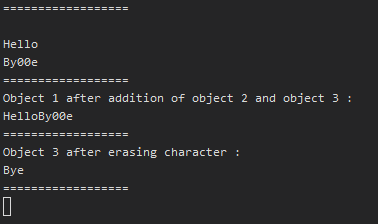
#pragma once  
  
#include <string>  
using namespace std;  
  
class cString  
{  
private:  
 char\* data;  
 int size;  
public:  
 int strSize();  
 void printString();  
 cString();  
 cString(char\*);  
 cString(const cString&);  
 cString operator +(cString&);  
 cString operator -(char);  
  
};

cString.cpp

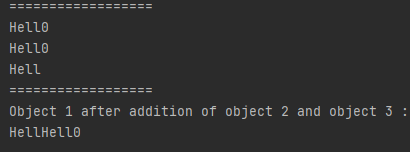
#include "cString.h"  
#include <iostream>  
#include<string>  
  
int cString::strSize() {  
 return size;  
}  
  
void cString::printString() {  
 for ( int i = 0; i < size; i++){  
 cout << data[i];  
 }  
 cout << endl;  
}  
  
cString::cString() {  
 this->data = new char[0];  
 size = 0;  
}  
  
cString::cString(char\* strInp) {  
 this->data = strInp;  
 size = strlen(strInp);  
}  
  
cString::cString(const cString &object) {  
 this->data = object.data;  
 this->size = object.size;  
}  
  
cString cString::operator+(cString& str) {  
 cString bufStr;  
 int indexer = 0;  
 bufStr.size = this->strSize() + str.strSize();  
 bufStr.data = new char[bufStr.size];  
 for (int i = 0; i < this->strSize(); i++)  
 {  
 bufStr.data[indexer] = this->data[i];  
 indexer++;  
 }  
 for (int i = 0; i < str.strSize(); i++)  
 {  
 bufStr.data[indexer] = str.data[i];  
 indexer++;  
 }  
 return bufStr;  
}  
  
cString cString::operator-(char chr) {  
 cString bufStr = cString();  
 bufStr.size = strSize()-1;  
 bufStr.data = new char[bufStr.size];  
 int indexer = 0;  
 for (int i = 0; i < this->strSize(); i++)  
 {  
 if (this->data[i] == chr) { continue; }  
 bufStr.data[indexer] = this->data[i];  
 indexer++;  
 }  
 return bufStr;  
}

1. **Екранні форми результату**

**C#**



**C++**



1. **Висновки**

Дана програма працює коректно та виконує поставлену задачу. Реалізовано на С# та C++.