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 №5 з дисципліни

**«Програмування - 2. Структури даних та алгоритми»**

**«ДОСЛІДЖЕННЯ МЕХАНІЗМУ УСПАДКУВАННЯ**

Виконав студент: ІС-03 Філоненко Андрій Ігорович

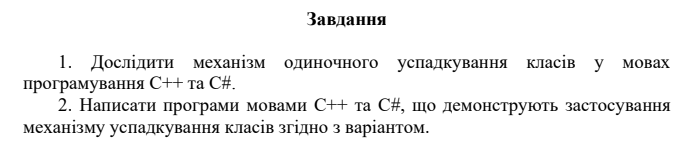
(шифр, прізвище, ім'я, по батькові)

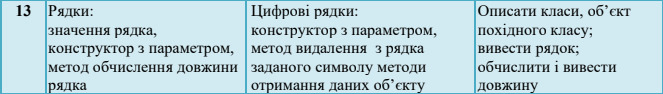
Перевірив викладач: Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2021

1. **Умова задачі**





1. **Текст файлів проекту**

Program.cs

using System;  
  
namespace Lab5\_OOP  
{  
 class Program  
 {  
 static void printStrings(string[] input)  
 {  
 for (int i = 0; i < input.Length; i++)  
 {  
 Console.WriteLine(input[i]);  
 }  
 }  
   
 static void Main(string[] args)  
 {  
 var obj = new dString("This", "Cont4ins", "444 Fours");  
  
 printStrings(obj.getData());  
  
 Console.WriteLine($"Length of line 1 : {obj.getsLength(0)}");  
 Console.WriteLine($"Length of line 2 : {obj.getsLength(1)}");  
 Console.WriteLine($"Length of line 3 : {obj.getsLength(2)}");  
  
 obj.removeDigit('4');  
   
 printStrings(obj.getData());  
   
 Console.WriteLine($"Length of line 2 : {obj.getsLength(1)}");  
 Console.WriteLine($"Length of line 3 : {obj.getsLength(2)}");  
 }  
 }  
}

BaseClass.cs

using System;  
  
namespace Lab5\_OOP  
{  
 internal class bString  
 {  
 protected string[] data { get; set; }  
  
 internal bString(params string[] input)  
 {  
 data = input;  
 }  
  
 public int getsLength(int index)  
 {  
 if ((index < 0) || (index > data.Length))  
 {  
 Console.WriteLine("Index out of range.");  
 return -1;  
 }  
  
 return data[index].Length;  
 }  
  
 public int getLength() => data.Length;  
 }  
}

}

DerivedClass.cs

using System;  
using System.Linq;  
  
namespace Lab5\_OOP  
{  
 internal class dString : bString  
 {  
 internal dString(params string[] input) : base(input)  
 {  
 data = input;  
 }  
   
 public string[] getData() => data;  
   
 public string this[int index]  
 {  
 get  
 {  
 try  
 {  
 return data[index];  
 }  
 catch (ArgumentOutOfRangeException)  
 {  
 Console.WriteLine("Index out of range.");  
 throw;  
 }  
 }  
 }  
  
 public void removeDigit(char digit)  
 {  
 for (int i = 0; i < data.GetLength(0); i++)  
 {  
 for (int j = 0; j < data[i].Length; j++)  
 {  
 if (data[i][j] == digit)  
 {  
 data[i] = data[i].Remove(j, 1);  
 j--;  
 }  
   
 }  
 }  
 }  
   
 }  
}

main.cpp

#include <iostream>  
#include"BaseClass.h"  
#include"DerivedClass.h"  
  
int main(){  
 dString object = dString(new string[3]{ "This", "Cont4ins", "Fours 44444"});  
  
 for(int i = 0; i < 3; i++){  
 cout << object.getData(i) << endl;  
 }  
 cout << endl;  
  
 cout << "Length of line 1 : " << object.getLength(0) << endl;  
 cout << "Length of line 2 : " << object.getLength(1) << endl;  
 cout << "Length of line 3 : " << object.getLength(2) << endl << endl;  
  
 object.removeDigit('4');  
 for(int i = 0; i < 3; i++){  
 cout << object.getData(i) << endl;  
 }  
  
 cout << endl << "Length of line 2 : " << object.getLength(1) << endl;  
 cout << "Length of line 3 : " << object.getLength(2) << endl << endl;  
}

BaseClass.h

#pragma once  
  
#include<string>  
  
using namespace std;  
  
class bString {  
protected:  
 string\* data;  
public:  
 bString(string[]);  
 int getLength(int);  
};

BaseClass.cpp

#include "BaseClass.h"  
#include<iostream>  
  
using namespace std;  
  
bString::bString(string input[]){  
 data = input;  
}  
  
int bString::getLength(int index){  
 if (index >= data->length() ){  
 cout << "Your index is out of range.";  
 return -1;  
 }  
 return data[index].length();  
}

DerivedClass.h

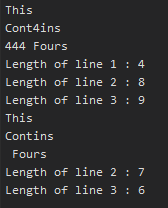
#pragma once  
#include"BaseClass.h"  
#include<iostream>  
  
using namespace std;  
  
class dString:  
 public bString  
{  
public:  
 dString(string[]);  
 string getData(int);  
 void removeDigit(char);  
};

DerivedClass.cpp

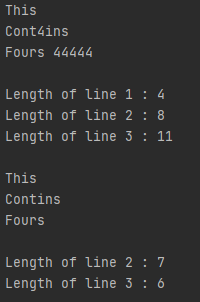
#include "DerivedClass.h"  
  
using namespace std;  
  
dString::dString(string input[]):  
bString(input) {  
 data = input;  
}  
  
string dString::getData(int index){  
 if (index >= data->length() ){  
 cout << "Your index is out of range.";  
 return " ";  
 }  
 return data[index];  
}  
  
void dString::removeDigit(char digit) {  
 for (int i = 0; i < data->length()-1; i++){  
 for (int j = 0; j < data[i].length(); j++){  
 if (data[i][j] == digit){  
 data[i].erase(j,1);  
 j--;  
 }  
 }  
 }  
}

1. **Екранні форми результату**

**C#**



**C++**



1. **Висновки**

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