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

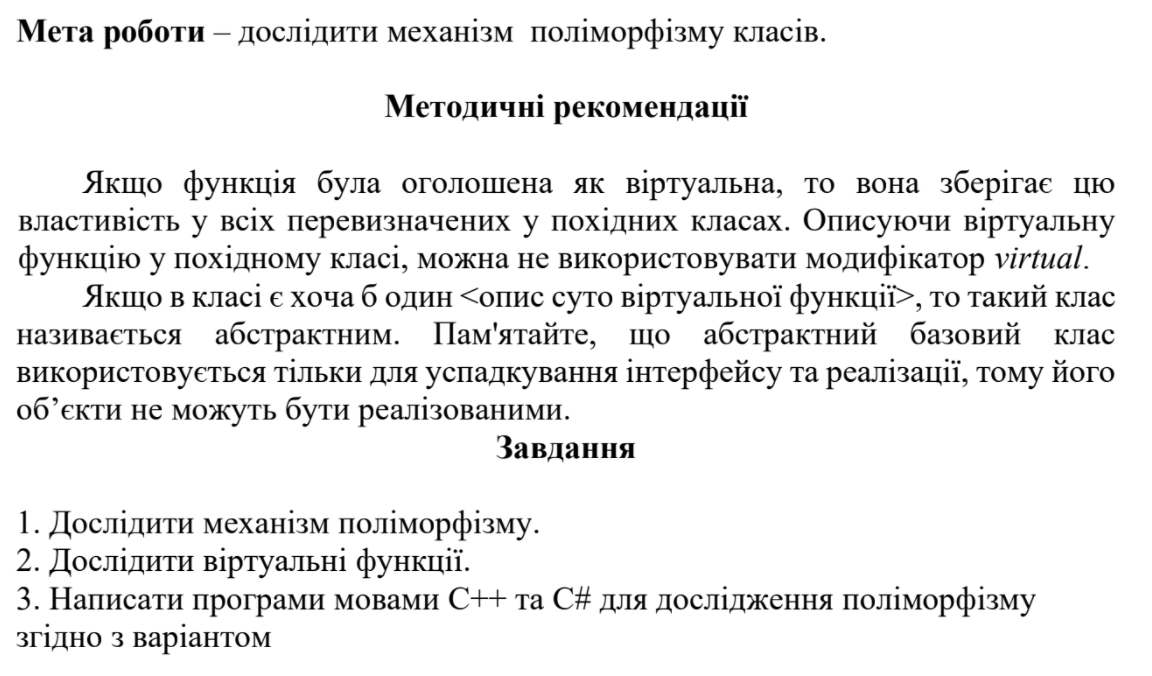
з лабораторної роботи № 6 на тему

«ДОСЛІДЖЕННЯ ПОЛІМОРФІЗМУ КЛАСІВ»

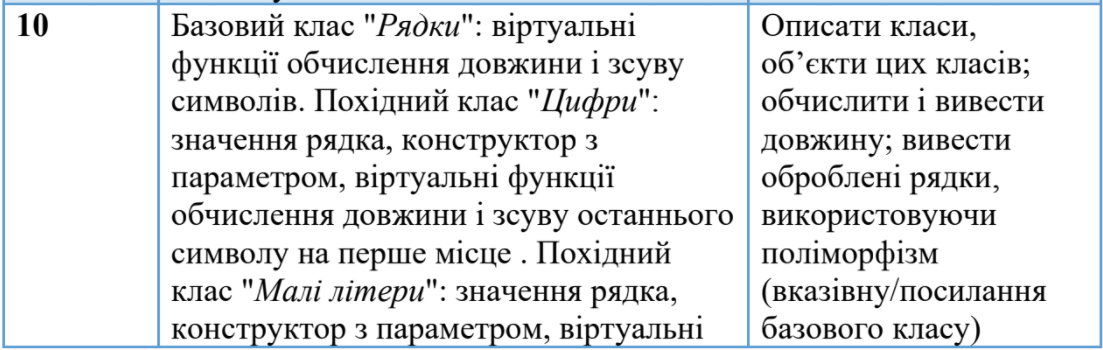
Виконав студент Кубай Д. І.

Київ 2021

**Варіант 10**



Варіант 10



**Код на мові С++**

**file “head.h”**

#pragma once

#include <iostream>

#include <string>

#include <cctype>

#include <algorithm>

using namespace std;

class Line

{

//line

protected:

string \_line;

public:

//line getter

string line()

{

return \_line;

}

//constructor

Line(string line)

{

\_line = line;

}

//length

virtual int Length()

{

return \_line.length();

}

//move the line

virtual void Move()

{

int n = \_line.length() - 1;

string new\_line;

for (int i = 0; i < \_line.length() - 1; i++)

{

\_line[i] = \_line[i + 1];

}

\_line[n] = '\0';

}

};

class Digits : public Line

{

public:

//constructor

Digits(string line) : Line(line) { }

//length

int Length() override

{

return Line::Length();

}

//move

void Move() override

{

int n = \_line.length();

char a = \_line[n - 1];

for (int i = \_line.length() - 1; i > 0 ; i--)

{

\_line[i] = \_line[i - 1];

}

\_line[0] = a;

}

//integer number from digits

int Number()

{

return stoi(\_line);

}

};

class SmallLetters : public Line

{

public:

//constructor

SmallLetters(string line) : Line(line)

{

\_line = line;

transform(\_line.begin(), \_line.end(), \_line.begin(),

[](unsigned char c)

{ return tolower(c); });

}

//length

int Length() override

{

return Line::Length();

}

//move

void Move() override

{

int n = \_line.length();

char a = \_line[0];

for (int i = 0; i < \_line.length() - 1; i++)

{

\_line[i] = \_line[i + 1];

}

\_line[n - 1] = a;

}

};

**file “Source.cpp”**

#include "head.h";

int main()

{

//input

cout << "Enter the line:\n";

string new\_line;

cin >> new\_line;

Line line\_1 = Line(new\_line);

cout << "Enter the digits:\n";

cin >> new\_line;

Digits digits\_1 = Digits(new\_line);

cout << "Enter the small letters:\n";

cin >> new\_line;

SmallLetters small\_1 = SmallLetters(new\_line);

//move the line

line\_1.Move();

digits\_1.Move();

small\_1.Move();

//output

cout << "\nline\_1 length: " << line\_1.Length();

cout << "\ndigits\_1 length: " << digits\_1.Length();

cout << "\nsmall\_1 length: " << small\_1.Length();

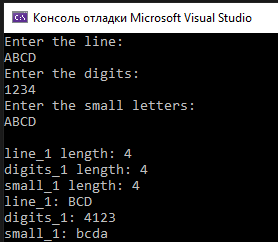
cout << "\nline\_1: " << line\_1.line();

cout << "\ndigits\_1: " << digits\_1.line();

cout << "\nsmall\_1: " << small\_1.line();

}

**Вивід програми (C++)**

****

**Код на мові С#**

**file “Program.cs”**

using System;

using LinesLibrary;

namespace Lab06

{

class Program

{

static void Main(string[] args)

{

//input

Console.WriteLine("Enter the line:");

string new\_line = Console.ReadLine();

Line line\_1 = new Line(new\_line);

Console.WriteLine("Enter the digits:");

new\_line = Console.ReadLine();

Line digits\_1 = new Digits(new\_line);

Console.WriteLine("Enter the small letters:");

new\_line = Console.ReadLine();

Line small\_1 = new SmallLetters(new\_line);

//move the line

line\_1.Move();

digits\_1.Move();

small\_1.Move();

//output

Console.WriteLine("\nline\_1 length: " + line\_1.Length());

Console.WriteLine("digits\_1 length: " + digits\_1.Length());

Console.WriteLine("small\_1 length: " + small\_1.Length());

Console.WriteLine("\nline\_1: " + line\_1.line);

Console.WriteLine("digits\_1: " + digits\_1.line);

Console.WriteLine("small\_1: " + small\_1.line);

}

}

}

**Library “LinesLibrary”**

**Class “Line.cs”**

using System;

namespace LinesLibrary

{

public class Line

{

//line

protected string \_line;

//line getter

public string line

{

get

{

return \_line;

}

}

//constructor

public Line(string line)

{

\_line = line;

}

//length

public virtual int Length()

{

return \_line.Length;

}

//move the line

public virtual void Move()

{

int n = \_line.Length - 1;

char[] new\_line = new char[n];

for (int i = 0; i < \_line.Length - 1; i++)

{

new\_line[i] = \_line[i + 1];

}

\_line = new string(new\_line);

}

}

}

**Class “Digits.cs”**

using System;

using System.Collections.Generic;

using System.Text;

namespace LinesLibrary

{

public class Digits : Line

{

//constructor

public Digits(string line) : base(line) { }

//length

public override int Length()

{

return base.Length();

}

//move

public override void Move()

{

int n = \_line.Length;

char[] new\_line = new char[n];

new\_line[0] = \_line[\_line.Length - 1];

for (int i = 1 ; i < \_line.Length; i++)

{

new\_line[i] = \_line[i - 1];

}

\_line = new string(new\_line);

}

//integer

public int Number()

{

return Convert.ToInt32(\_line);

}

}

}

**Class “SmallLetters.cs”**

using System;

using System.Collections.Generic;

using System.Text;

namespace LinesLibrary

{

public class SmallLetters : Line

{

//constructor

public SmallLetters(string line) : base(line)

{

\_line = line.ToLower();

}

//length

public override int Length()

{

return base.Length();

}

//move

public override void Move()

{

int n = \_line.Length;

char[] new\_line = new char[n];

for (int i = 0; i < \_line.Length - 1; i++)

{

new\_line[i] = \_line[i + 1];

}

new\_line[n - 1] = \_line[0];

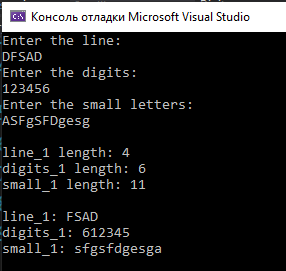
\_line = new string(new\_line);

}

}

}

**Вивід програми (C#)**

****

**Висновок**

Я досліждував поліморфізм. При наслідування віртуальні функції можна переопреділити, щоб вона відповідала класу нащадку.

Я створив 2 програми на C# і C++, які мають базовий клас рядків і два унаслідувані від нього класи: один з цифрами, а другий з малими літерами. У базовому класі є віртуальна функція зсуву, яка буде у кожному класові – наслідникові перевизначеною і працювати по іншому.