

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет

ЗВІТ  
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 12  
з навчальної дисципліни  
“Базові методології та технології програмування”

Програмна реалізація абстрактних типів даних

ЗАВДАННЯ ВИДАВ  
доцент кафедри кібербезпеки  
та програмного забезпечення  
Доренський О. П.  
<https://github.com/odorenskyi/>

ВИКОНАВ  
студент академічної групи  
КІ22-2  
Ткаченко О. Ю.

ПЕРЕВІРИВ  
ст. викладач  
кафедри кібербезпеки  
та програмного забезпечення  
Собінов О. Г.

**Мета:** Набуття ґрунтовних вмінь і практичних навичок об’єктного аналізу й проектування, створення класів С++ та тестування їх екземплярів, використання препроцесорних директив, макросів і макрооператорів під час реалізації програмних засобів у кросплатформовому середовищі Code::Blocks.

### **Варіант №36**

**Завдання:** 1. Як складову заголовкового файлу ModulesПрізвище.h розробити клас ClassLab12\_Прізвище — формальне представлення абстракції сутності предметної області (об’єкта) за варіантом, — поведінка об’єкта якого реалізовує розв’язування задачі 12.1.

2. Реалізувати додаток Teacher, який видає 100 звукових сигналів і в текстовий файл TestResults.txt записує рядок “Встановлені вимоги порядку виконання лабораторної роботи порушено!”, якщо файл проектуа main.cpp під час його компіляції знаходився не в \Lab12\prj, інакше — створює об’єкт класу ClassLab12\_Прізвище із заголовкового файлу ModulesПрізвище.h та виконує його unit-тестування за тест-сьютом(ами) із \Lab12\TestSuite\, протоколюючи результати тестування в текстовий файл \Lab12\TestSuite\TestResults.txt.

Опис класу Class12\_Tkachenko знаходиться в заголовному файлі Class12\_Tkachenko\_Header.h, лістинг якого знаходиться в Додатку А.

Імплементація класу Class12\_Tkachenko знаходиться в бібліотечному файлі Class12\_Tkachenko\_Module.a, лістинг якого знаходиться в Додатку Б.

Лістинг додатку Teacher.exe знаходиться в Додатку В цього звіту.

Тест сьют, за яким виконуються юніт тести знаходиться в Додатку Д.

Вміст файлу TestResults:

unit test #1: passed

unit test #2: failed

unit test #3: failed

unit test #4: passed

Загалом статус тестування: **passed**

**Висновок:** Виконавши цю лабораторну роботу з теми “Програмна реалізація абстрактних типів даних” я набув ґрунтовних вмінь і практичних навичок об’єктного аналізу й проектування, створення класів C++ та тестування їх екземплярів, використання препроцесорних директив, макросів і макрооператорів під час реалізації програмних засобів у кросплатформовому середовищі Code::Blocks.

Програмна реалізація абстрактних типів даних може впливати на продуктивність програми. Оптимальна реалізація дозволяє забезпечити швидкий доступ до даних та виконання операцій над ними. Розробники повинні уважно вибирати алгоритми та структури даних для досягнення найкращої продуктивності.

Правильна реалізація абстрактних типів даних є ключовим аспектом забезпечення надійності програмного забезпечення. Недостатньо лише правильно розробити алгоритми; також потрібно ретельно управляти пам'яттю та уникати можливих проблем, таких як витоки пам'яті або неправильна робота з покажчиками.

Добре розроблені абстрактні типи даних можуть бути легко перевикористані в різних програмах. Це дозволяє економити час та зусилля розробників, а також покращує якість програмного забезпечення.

Якщо програмна реалізація абстрактних типів даних добре структурована та документована, це сприяє полегшенню розробки та обслуговування програм. Розробники можуть швидше зрозуміти, як працює реалізація та які операції можна виконувати з використанням цих типів даних.

## Додаток А

### Лістинг Class12\_Tkachenko\_Header.h

```
#ifndef CLASS12_TKACHENKO_HEADER_H_INCLUDED
#define CLASS12_TKACHENKO_HEADER_H_INCLUDED
#include <string>
struct attributes{
    int mmHeight = 0, mmWidth = 0;
    std::string boardText;
};

class Class12_Tkachenko{
public:
    int mmReturnSquare();
    int setSize(std::string type, int mmNewSize);
    int setSize(int mmNewHeight, int mmNewWidth);

    void changeText(std::string newText="Тут могла б бути ваша реклама");

    attributes returnAttributes(std::string type = "all");
    attributes returnAttributes(std::string type1, std::string type2);

    Class12_Tkachenko(int mmWidth, int mmHeight, std::string boardText =
"Тут могла б бути ваша реклама" );
private:
    int width;
    int height;

    std::string text;
};
```

```
#endif // CLASS12_TKACHENKO_HEADER_H_INCLUDED
```

## **Додаток Б**

### **Лістинг Class12\_Tkachenko\_Module.a**

```
#include <string>
```

```
#include <Class12_Tkachenko_Header.h>
```

```
Class12_Tkachenko::Class12_Tkachenko(int mmWidth, int mmHeight, std::string  
boardText){
```

```
    height = mmHeight;
```

```
    width = mmWidth;
```

```
    text = boardText;
```

```
};
```

```
int Class12_Tkachenko::mmReturnSquare(){
```

```
    return width * height;
```

```
};
```

```
void Class12_Tkachenko::changeText(std::string newText){
```

```
    text = newText;
```

```
    return;
```

```
};
```

```
int Class12_Tkachenko::setSize(int mmNewHeight, int mmNewWidth){
```

```
    if (mmNewHeight > 0 && mmNewWidth > 0 ){
```

```
        width = mmNewWidth;
```

```
        height = mmNewHeight;
```

```
        return 1;
```

```
    }
```

```
    return 0;
```

```
};
```

```
int Class12_Tkachenko::setSize(std::string type, int mmNewSize){  
    if (mmNewSize > 0){  
        if(type == "width"){  
            width = mmNewSize;  
        } else if(type == "height"){  
            height = mmNewSize;  
        } else {  
            return 0;  
        }  
        return 1;  
    }  
    return 0;  
}
```

```
attributes Class12_Tkachenko::returnAttributes(std::string type){  
    attributes atrLst;  
    if(type == "all"){  
        atrLst.mmWidth = width;  
        atrLst.mmHeight = height;  
        atrLst.boardText = text;  
    } else if (type == "width"){  
        atrLst.mmWidth = width;  
    } else if (type == "height"){  
        atrLst.mmHeight = height;  
    } else if (type == "text"){  
        atrLst.boardText = text;  
    }  
    return atrLst;  
}
```

```
};
attributes Class12_Tkachenko::returnAttributes(std::string type1, std::string
type2){
    attributes atrLst;
    if (type1 == "width" || type2 == "width"){atrLst.mmWidth = width;}
    if (type1 == "height" || type2 == "height" ){atrLst.mmHeight = height;}
    if (type1 == "text" || type2 == "text"){atrLst.boardText = text;}
    return atrLst;
};
```

## **Додаток В**

### **Лістинг Teacher.exe**

```
#include <iostream>
#include <fstream>
#include <filesystem>
#include <Class12_Tkachenko_Header.h>

using namespace std;
int main()
{
    system("chcp 65001");
    system("cls");
    ofstream testResults("../TestSuite/TestResults.txt", ios_base::out|ios::binary);
    try{
        if(!(testResults.is_open())){
            throw(1);
        }
    } catch (int exc){
        cout<<"Не вдалось відкрити файл...";
        getchar();
        return 0;
    }
```

```
}
```

```
string curDir = __FILE__;  
if (curDir.find("\\lab12\\prj") == string::npos){  
    for(int i = 0; i<100; i++){  
        cout<<"\a";  
    }  
} else{  
    ifstream unit("../TestSuite/unitTests.txt", ios_base::in|ios::binary);  
    try{  
        if (!(unit.is_open())){throw(1);}  
    } catch(int exc){  
        cout<< "Файлу unitTests.txt не існує...";  
        getchar();  
        return 0;  
    }  
};
```

```
string line;  
int elemNumber = 0;  
getline(unit, line);  
Class12_Tkachenko newObject(0,0);  
attributes atrLst;  
int gotRight;  
while(getline(unit, line, '\t')){  
    switch(elemNumber%4){  
        case 0:  
            gotRight = 0;  
            if(newObject.setSize("width", stoi(line))){  
                atrLst = newObject.returnAttributes();  
            }  
    }  
}
```



```

        if (atrLst.mmWidth == stoi(line)){
            gotRight++;
        }
    };
    break;
case 1:
    if(newObject.setSize("height", stoi(line))){
        atrLst = newObject.returnAttributes();
        if (atrLst.mmHeight == stoi(line)){
            gotRight++;
        }
    };
    break;
case 2:
    newObject.changeText(line);
    atrLst = newObject.returnAttributes();
    if (atrLst.boardText == line){
        gotRight++;
    }
    break;
case 3:
    if (newObject.mmReturnSquare() == stoi(line)){
        gotRight++;
    }
    if (gotRight == 4){
        testResults << "unit test #" <<(elemNumber+1)/4<<": passed\n";
    } else{
        testResults << "unit test #" <<(elemNumber+1)/4<<": failed\n";
    }
    break;

```

```

    }
    elemNumber++;
}
return 0;
}
}

```

## Додаток Д

### Тест сьют

width	height	text	square	expected results
10	10	Тут могла б бути ваша реклама	100	passed
0	1	Тут могла б бути ваша реклама	0	failed
1	0	Тут могла б бути ваша реклама	0	failed
10000100000		Завітайте на виставу завтра!	1000000000	passed