# **Звіт про виконання лабораторної роботи № 1.7**

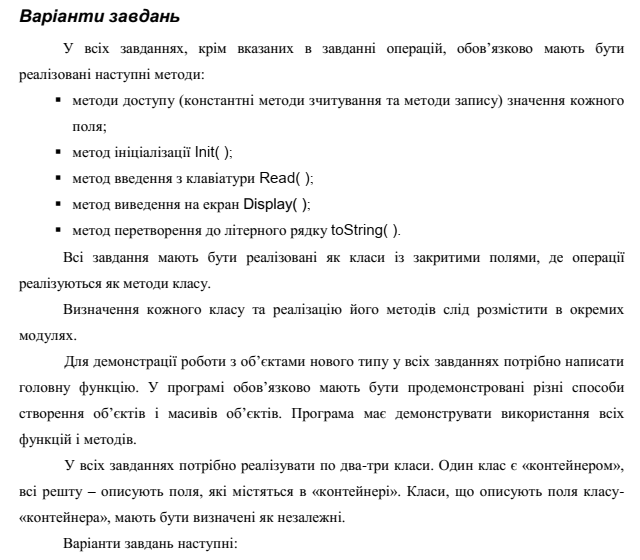
## «Композиція класів та об’єктів – складніші завдання» з дисципліни «Об’єктно-орієнтоване програмування»

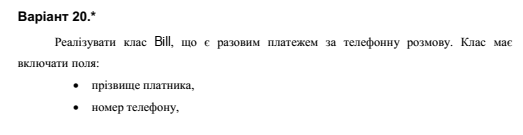
## Студента групи «**ІТ-12**» - **Степанчука Сергія**

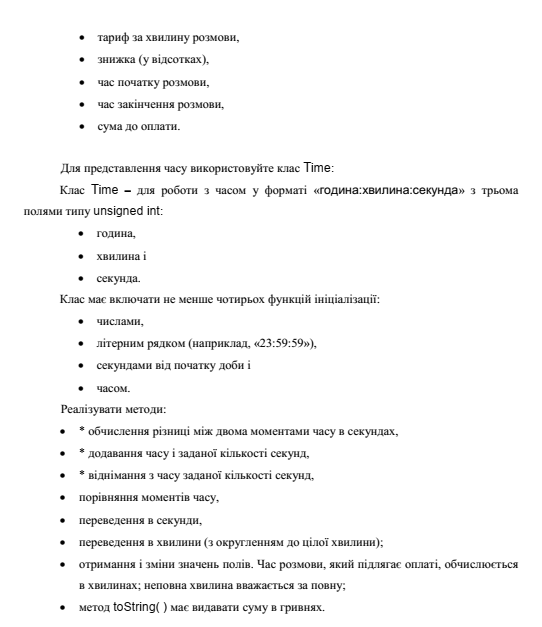
## Мета роботи

Освоїти використання композитних класів та об’єктів.

## Умова завдання



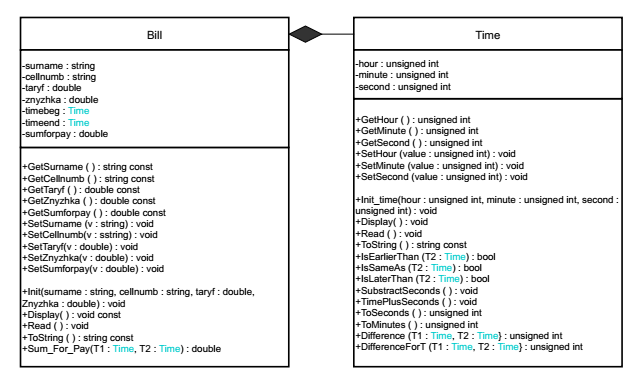




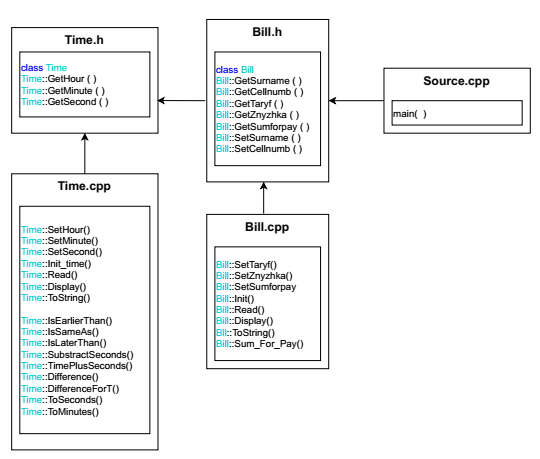
## Посилання на git-репозиторій з проектом:

<https://github.com/SergiyStepanchuk/OOP_Lab_1.7>

## UML-діаграма класів



## Структурна схема



## Текст програми

### // Bill.h

#pragma once

#include <iostream>

#include <sstream>

#include <string>

#include "Time.h"

using namespace std;

class Bill

{

private:

string surname;

string cellnumb;

double taryf;

double znyzhka;

Time timebeg;

Time timeend;

double sumforpay;

public:

string GetSurname() const { return surname; };

void SetSurname(string v) { this->surname = v; };

string GetCellnumb() const { return cellnumb; };

void SetCellnumb(string v) { this->cellnumb = v; }

double GetTaryf() const { return taryf; };

void SetTaryf(double);

double GetZnyzhka() const { return znyzhka; };

void SetZnyzhka(double);

double GetSumforpay() const { return sumforpay; };

void SetSumforpay(double);

void Read();

void Display();

void Init(string, string, double, double);

string toString() const;

double Sum\_For\_Pay(Time T1, Time T2);

};

### // Bill.cpp

#include "Bill.h"

void Bill::SetTaryf(double v)

{

this->taryf = v;

}

void Bill::SetZnyzhka(double v)

{

if (v > 100 || v < 0)

{

v = 0;

}

this->znyzhka = v;

}

void Bill::SetSumforpay(double v)

{

this->sumforpay = v;

}

void Bill::Init(string surname, string cellnumb, double taryf, double znyzhka)

{

SetSurname(surname);

SetCellnumb(cellnumb);

SetTaryf(taryf);

SetZnyzhka(znyzhka);

}

void Bill::Read()

{

string surname;

cout << " Enter surname: "; cin >> surname;

string cellnumb;

cout << " Enter phone number: "; cin >> cellnumb;

double taryf, znyzhka;

cout << " Enter taryf per minute: "; cin >> taryf;

do { cout << " Enter discount, %: "; cin >> znyzhka;

} while (znyzhka > 100 || znyzhka < 0);

Init(surname, cellnumb, taryf, znyzhka);

}

string Bill::toString() const

{

stringstream sout;

sout << endl << " Surname: " << surname << endl;

sout << " Phone number: " << cellnumb << endl;

sout << " Taryf per minute: " << taryf << "UAH" << endl;

sout << " Discount: " << znyzhka << "%" << endl;

return sout.str();

}

void Bill::Display()

{

cout << toString() << endl;

}

double Bill::Sum\_For\_Pay(Time T1, Time T2)

{

return DifferenceForT(T1,T2) \* taryf - (DifferenceForT(T1, T2) \* taryf) \* (znyzhka / 100);

}

### // Time.h

#pragma once

#include <iostream>

#include <sstream>

#include <string>

using namespace std;

class Time

{

private:

unsigned int hour, minute, second;

public:

unsigned int GetHour() const { return hour; };

unsigned int GetMinute() const { return minute; };

unsigned int GetSecond() const { return second; };

void SetHour(unsigned int);

void SetMinute(unsigned int);

void SetSecond(unsigned int);

void Init\_time(unsigned int, unsigned int, unsigned int);

void Read();

void Display();

string toString() const;

//////////////////////////////////////////////////////////////////////

// Порівняння моментів часу

bool IsEarlierThan(Time);

bool IsSameAs(Time);

bool IsLaterThan(Time);

//відніманняння з часу заданої кількості секунд

void SubstractSeconds();// хз, чи правильно

//додавання до часу задану кількість секунд

void TimePlusSeconds();// хз, чи правильно

//обчислення різниці між двома моментами часу в секундах

friend unsigned int Difference(Time, Time);

friend unsigned int DifferenceForT(Time T1, Time T2);//minutes

// Переведення в секунди

unsigned int ToSeconds();

// Переведення у хвилини (з округленням до цілої хвилини)

unsigned int ToMinutes();

};

### // Time.cpp

#include "Time.h"

#include <iostream>

void Time::SetHour(unsigned int value)

{

if (value > 23 || value < 0)

value = 0;

this->hour = value;

}

void Time::SetMinute(unsigned int value)

{

if (value > 59 || value < 0)

value = 0;

this->minute = value;

}

void Time::SetSecond(unsigned int value)

{

if (value > 59 || value < 0)

value = 0;

this->second = value;

}

void Time::Init\_time(unsigned int hour, unsigned int minute, unsigned int second)

{

SetHour(hour);

SetMinute(minute);

SetSecond(second);

}

void Time::Read()

{

unsigned int h, m, s;

do {

cout << " Enter hour: "; cin >> h;

} while (h > 23 || h < 0);

do {

cout << " Enter minute: "; cin >> m;

} while (m > 59 || m < 0);

do {

cout << " Enter second: "; cin >> s;

} while (s > 59 || s < 0);

Init\_time(h, m, s);

}

void Time::Display() {

cout << endl << toString() << endl;

}

string Time::toString() const {

stringstream sout;

sout << " " << hour << ":" << minute << ":" << second;

return sout.str();

}

////////////////////////////////////////////////////////////////////////

bool Time::IsEarlierThan(Time T2)

{

if (hour == T2.hour)

{

if (minute == T2.minute)

{

if (second == T2.second)

return false;

else if (second < T2.second)

return true;

else

return false;

}

else if (minute < T2.minute)

return true;

else

return false;

}

else if (hour < T2.hour)

return true;

else

return false;

}

bool Time::IsSameAs(Time T2)

{

if (hour == T2.hour)

{

if (minute == T2.minute)

{

if (second == T2.second)

return true;

else

return false;

}

else

return false;

}

else

return false;

}

bool Time::IsLaterThan(Time T2)

{

if (IsEarlierThan(T2))

return false;

else if (!IsSameAs(T2))

return true;

}

void Time::SubstractSeconds()

{

unsigned int substr;

do {

cout << "Enter how much seconds to substract: "; cin >> substr;

} while (substr < 1);

hour -= trunc(substr / 3600.);

minute -= trunc(substr % 3600 / 60.);

second -= substr % 60;

if (second >= 60 || second < 0)

{

minute += second / 60;

second %= 60;

if (second < 0)

{

minute--;

second = 60 + second;

}

}

if (minute >= 60 || minute < 0)

{

hour += minute / 60;

minute = minute % 60;

if (minute < 0)

{

hour--;

minute = 60 + minute;

}

}

if (hour >= 24 || hour < 0)

hour = (hour < 0 ? 24 : 0) + hour % 24;

}

void Time::TimePlusSeconds()

{

unsigned int seconds;

do {

cout << "Enter how much seconds to add: "; cin >> seconds;

} while (seconds < 1);

int hm = trunc(seconds / 3600.);

hour += trunc(seconds / 3600.);

minute += trunc(seconds % 3600 / 60.);

second += seconds % 60;

if (second >= 60 || second < 0)

{

minute += second / 60;

second %= 60;

if (second < 0)

{

minute--;

second = 60 + second;

}

}

if (minute >= 60 || minute < 0)

{

hour += minute / 60;

minute = minute % 60;

if (minute < 0)

{

hour--;

minute = 60 + minute;

}

}

if (hour >= 24 || hour < 0)

hour = (hour < 0 ? 24 : 0) + hour % 24;

}

unsigned int Time::ToSeconds()

{

unsigned int t;

t = hour \* 3600 + minute \* 60 + second;

return t;

}

unsigned int Time::ToMinutes()

{

unsigned int t;

if (second < 1)

{

t = hour \* 60 + minute;

}

if (second > 0)

{

t = hour \* 60 + minute + 1;

}

return t;

}

unsigned int Difference(Time T1, Time T2)

{

return T2.ToSeconds() - T1.ToSeconds();

}

unsigned int DifferenceForT(Time T1, Time T2)

{

int k;

k = Difference(T1, T2) / 60.;

return k;

}

### // main.cpp

#include <iostream>

#include "Time.h"

#include "Bill.h"

using namespace std;

int main()

{

Bill B1;

Time T1, T2; //T1-timebeg, T2 - timeend

B1.Read();

cout << endl;

cout << " Enter talk start time:" << endl;

T1.Read();

cout << " Enter talk end time:" << endl;

T2.Read();

cout << "------------------------------------------------------------------------";

B1.Display();

cout << " Talk start time: ";

T1.Display();

cout << " Talk end time: ";

T2.Display();

cout << endl;

cout << "Sum to pay: " << B1.Sum\_For\_Pay(T1, T2) << "UAH" << endl;

cout << "------------------------------------------------------------------------" << endl;

cout << " MANIPULATIONS WITH TIME " << endl << endl;

cout << "Talk start time is earlier than talk end time: ";

if (T1.IsEarlierThan(T2))

cout << " TRUE" << endl << endl;

else

cout << " FALSE" << endl << endl;

cout << "Talk end time minus talk start time (in seconds): ";

unsigned int n = Difference(T1, T2);

cout << n << endl;

cout << "Talk end time minus talk start time (in minutes): ";

unsigned int k = DifferenceForT(T1, T2);

cout << k << endl << endl;

cout << "Talk start time in seconds: ";

unsigned int s = T1.ToSeconds();

cout << s << endl;

cout << "Talk start time in minutes: ";

unsigned int m = T1.ToMinutes();

cout << m << endl << endl;

cout << "Subtracting from time a specified number of seconds (talk start time): " << endl;

T1.SubstractSeconds();

T1.Display();//

cout << endl;

cout << "Adding to time a specified number of seconds (talk start time which was substracted): " << endl;

T1.TimePlusSeconds();

T1.Display();//

return 0;

}

## UnitTest

### Код

#include "pch.h"

#include "CppUnitTest.h"

#include "../Program/Bill.h"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest17

{

TEST\_CLASS(UnitTest17)

{

public:

TEST\_METHOD(TestMethod1)

{

Bill B1;

B1.SetSurname("Stepanchuk");

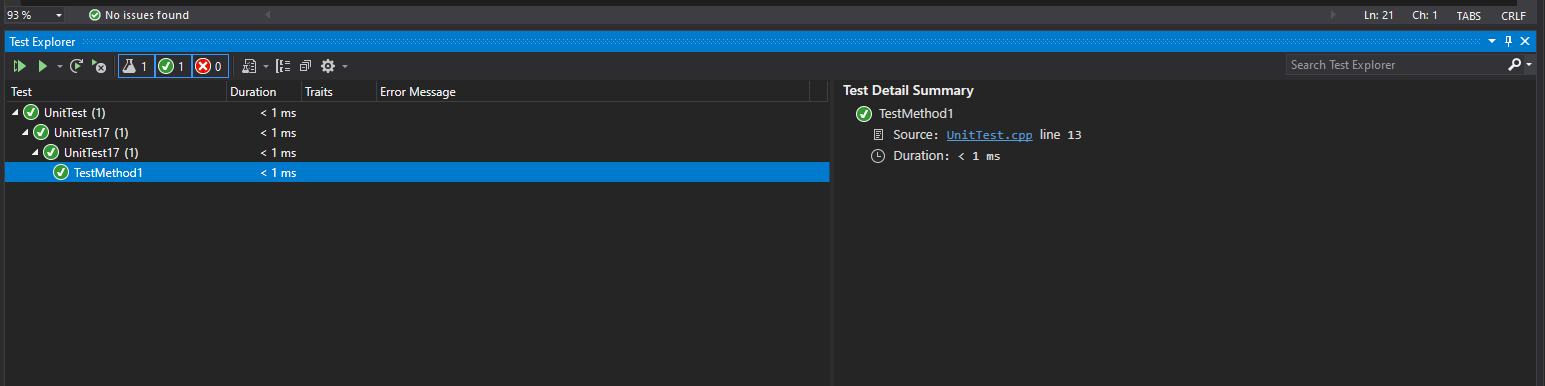
Assert::AreEqual("Stepanchuk", B1.GetSurname().c\_str());

}

};

}

### Результат



## Висновок

Навчився використання композитних класів та об’єктів.