Задача 3. Створити клас TPair, який представляє пару чисел і містить методи для їх збільшення/зменшення на 1. Реалізувати класи нащадки TTime (“години.хвилини”) та TMoney (“гривні.копійки”). Згенерувати поступово випадковим чином  пар (час, гроші), де час – тривалість виконання роботи, а гроші – вартість однієї хвилини роботи працівників. Обчислити витрати на виконання кожної із робіт.

----------------- RangeNumber.h ------------------

#pragma once

#include<climits>

class RangeNumber

{

int minValue;

int maxValue;

int value;

void set\_minValue(int value);

int get\_minValue();

void set\_maxValue(int value);

int get\_maxValue();

public:

void set\_value(int value);

int get\_value();

RangeNumber(int minValue, int maxValue, int value);

RangeNumber(int minValue, int maxValue);

RangeNumber(int minValue);

RangeNumber();

bool increase();

bool decrease();

};

----------------- RangeNumber.cpp ------------------

#include "stdafx.h"

#include "RangeNumber.h"

void RangeNumber::set\_minValue(int value)

{

minValue = value;

}

int RangeNumber::get\_minValue()

{

return minValue;

}

void RangeNumber::set\_maxValue(int value)

{

if (value < minValue)

throw "Value is lower than min";

maxValue = value;

}

int RangeNumber::get\_maxValue()

{

return maxValue;

}

void RangeNumber::set\_value(int value)

{

if (value < minValue || value > maxValue)

throw "Wrong number";

this->value = value;

}

int RangeNumber::get\_value()

{

return value;

}

RangeNumber::RangeNumber(int minValue, int maxValue, int value)

{

set\_minValue(minValue);

set\_maxValue(maxValue);

set\_value(value);

}

RangeNumber::RangeNumber(int minValue, int maxValue):RangeNumber(minValue, maxValue, minValue)

{

}

RangeNumber::RangeNumber(int minValue):RangeNumber(minValue, INT\_MAX)

{

}

RangeNumber::RangeNumber():RangeNumber(INT\_MIN)

{

}

bool RangeNumber::increase()

{

value++;

if (value <= maxValue)

return true;

else

{

value = minValue;

return false;

}

}

bool RangeNumber::decrease()

{

value--;

if (value >= minValue)

return true;

else

{

value = maxValue;

return false;

}

}

----------------------- Pair.h--------------------

#pragma once

#include "RangeNumber.h"

class Pair

{

public:

RangeNumber\* number1;

RangeNumber\* number2;

Pair(int minV1, int maxV1, int value1,

int minV2, int maxV2, int value2);

Pair(int minV1, int maxV1, int minV2, int maxV2);

Pair(int minV1, int minV2);

Pair();

void increase();

void decrease();

};

----------------------- Pair.cpp--------------------

#include "stdafx.h"

#include "Pair.h"

Pair::Pair(int minV1, int maxV1, int value1,

int minV2, int maxV2, int value2)

{

number1 = new RangeNumber(minV1, maxV1, value1);

number2 = new RangeNumber(minV2, maxV2, value2);

}

Pair::Pair(int minV1, int maxV1, int minV2, int maxV2)

{

number1 = new RangeNumber(minV1, maxV1);

number2 = new RangeNumber(minV2, maxV2);

}

Pair::Pair(int minV1, int minV2)

{

number1 = new RangeNumber(minV1);

number2 = new RangeNumber(minV2);

}

Pair::Pair()

{

number1 = new RangeNumber();

number2 = new RangeNumber();

}

void Pair::increase()

{

if (!number2->increase())

{

number1->increase();

}

}

void Pair::decrease()

{

if (!number2->decrease())

{

number1->decrease();

}

}

----------------- MyTime.h-----------------------

#pragma once

#include "Pair.h"

class MyTime :

protected Pair

{

public:

void set\_hour(int value);

int get\_hour();

void set\_minute(int value);

int get\_minute();

MyTime(int hours, int minutes);

MyTime();

using Pair::increase;

using Pair::decrease;

};

----------------- MyTime.cpp-----------------------

#include "stdafx.h"

#include "MyTime.h"

void MyTime::set\_hour(int value)

{

number1->set\_value(value);

}

int MyTime::get\_hour()

{

return number1->get\_value();

}

void MyTime::set\_minute(int value)

{

number2->set\_value(value);

}

int MyTime::get\_minute()

{

return number2->get\_value;

}

MyTime::MyTime(int hours, int minutes):

Pair(0,23,hours,0,59,minutes)

{

}

MyTime::MyTime():MyTime(0,0)

{

}

------------------- MyMoney.h--------------------------

#pragma once

#include "Pair.h"

class MyMoney :

protected Pair

{

public:

void set\_grivnas(int value);

int get\_grivnas();

void set\_coins(int value);

int get\_coins();

MyMoney(int grivnas, int coins);

MyMoney();

using Pair::increase;

using Pair::decrease;

};

------------------- MyMoney.cpp--------------------------

#include "stdafx.h"

#include "MyMoney.h"

void MyMoney::set\_grivnas(int value)

{

number1->set\_value(value);

}

int MyMoney::get\_grivnas()

{

return number1->get\_value();

}

void MyMoney::set\_coins(int value)

{

number2->set\_value(value);

}

int MyMoney::get\_coins()

{

return number2->get\_value();

}

MyMoney::MyMoney(int grivnas, int coins):

Pair(0, INT\_MAX, grivnas, 0, 99, coins)

{

}

MyMoney::MyMoney():MyMoney(0,0)

{

}

---------------------------- Order.h--------------

**ЗАКІНЧИТИ**

#pragma once

#include "MyTime.h"

#include "MyMoney.h"

class Order

{

public:

MyTime\* time;

MyMoney\* money;

};