#pragma once

#include<string>

using namespace std;

class PairValue

{

int \_Value;

int \_MinValue;

int \_MaxValue;

public:

int getValue();

void setValue(int value);

int getMinValue();

void setMinValue(int value);

int getMaxValue();

void setMaxValue(int value);

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

PairValue(int value, int minValue);

PairValue(int value);

PairValue();

bool isCorrectValue(int value);

int getRageCount(int value);

string toString();

};

//------------------------------------

#include "stdafx.h"

#include "PairValue.h"

#include <string>

using namespace std;

bool PairValue::isCorrectValue(int value)

{

return value>=\_MinValue && value<\_MaxValue;

}

int PairValue::getRageCount(int value)

{

if (value<\_MinValue)

{

throw "the value is incorrect";

}

return (value-\_MinValue)/(\_MaxValue-\_MinValue);

}

string PairValue::toString()

{

return to\_string(\_Value);

}

int PairValue::getValue()

{

return \_Value;

}

void PairValue::setValue(int value)

{

if (isCorrectValue(value))

{

\_Value = value;

}

else

{

throw "the value is incorrect";

}

}

int PairValue::getMinValue()

{

return \_MinValue;

}

void PairValue::setMinValue(int value)

{

\_MinValue = value;

}

int PairValue::getMaxValue()

{

return \_MaxValue;

}

void PairValue::setMaxValue(int value)

{

if (value<\_MinValue)

{

throw "the value is incorrect";

}

\_MaxValue = value;

}

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

{

setMinValue(minValue);

setMaxValue(maxValue);

setValue(value);

}

PairValue::PairValue(int value, int minValue):PairValue(value,minValue, \_I32\_MAX)

{

}

PairValue::PairValue(int value):PairValue(value,0)

{

}

PairValue::PairValue():PairValue(0,\_I32\_MIN,\_I32\_MAX)

{

}

//=====================================

#pragma once

#include"PairValue.h"

#include <string>

using namespace std;

class Pair

{

PairValue\* \_value1;

PairValue\* \_value2;

public:

int getValue1();

void setValue1(int value);

int getValue2();

void setValue2(int value);

Pair(int value1, int minValue1,int maxValue1, int value2, int minValue2, int maxValue2);

Pair(int minValue1, int maxValue1, int minValue2, int maxValue2);

Pair();

string toString(char separator='.');

};

//--------------------------------------

#include "stdafx.h"

#include "Pair.h"

#include <string>

using namespace std;

int Pair::getValue1()

{

return \_value1->getValue();

}

void Pair::setValue1(int value)

{

\_value1->setValue(value);

}

int Pair::getValue2()

{

return \_value2->getValue();

}

void Pair::setValue2(int value)

{

\_value2->setValue(value);

}

Pair::Pair(int value1, int minValue1, int maxValue1, int value2, int minValue2, int maxValue2)

{

\_value1 = new PairValue(value1, minValue1, maxValue1);

\_value2 = new PairValue(value2, minValue2, maxValue2);

}

Pair::Pair(int minValue1, int maxValue1, int minValue2, int maxValue2):

Pair(minValue1,minValue1,maxValue1,minValue2,minValue2,maxValue2)

{

}

Pair::Pair():Pair(\_I32\_MIN,\_I32\_MAX, \_I32\_MIN, \_I32\_MAX)

{}

string Pair::toString(char separator)

{

return \_value2->toString() + separator + \_value1->toString();

}

//==========================================

#pragma once

#include "Pair.h"

#include <string>

using namespace std;

class Time :

protected Pair

{

public:

int getMinutes();

void setMinutes(int value);

int getHours();

void setHours(int value);

Time(int hours, int minutes);

Time();

string toString();

};

//------------------------------------

#include "stdafx.h"

#include "Time.h"

#include <string>

using namespace std;

int Time::getMinutes()

{

return getValue1();

}

void Time::setMinutes(int value)

{

setValue1(value);

}

int Time::getHours()

{

return getValue2();

}

void Time::setHours(int value)

{

setValue2(value);

}

Time::Time(int hours,int minutes):Pair(minutes,0,60,hours,0,24)

{

}

Time::Time():Time(0,0)

{

}

string Time::toString()

{

return Pair::toString(':');

}

//===========================================

// ConsoleApplication88.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include"Time.h"

#include <iostream>

#include <string>

using namespace std;

int main()

{

Time t1(3, 23);

cout << t1.toString() << endl;

system("pause");

return 0;

}