Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Информатика»

"Классы и объекты.Шаблоны классов ."

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Бадртдинов Т.З

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2023

# Постановка задачи

1. Определить шаблон класса-контейнера (см. лабораторную работу №6). 2. Реализовать конструкторы, деструктор, операции ввода-вывода, операцию присваивания.

3. Перегрузить операции, указанные в варианте.

4. Инстанцировать шаблон для стандартных типов данных (int, float, double).

5. Написать тестирующую программу, иллюстрирующую выполнение операций для контейнера, содержащего элементы стандартных типов данных.

6. Реализовать пользовательский класс (см. лабораторную работу №3).

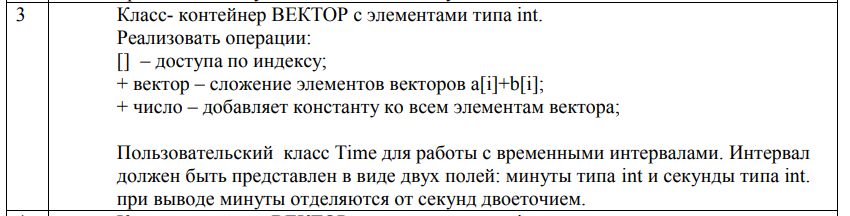
7. Перегрузить для пользовательского класса операции ввода-вывода.

8. Перегрузить операции необходимые для выполнения операций контейнерного класса.

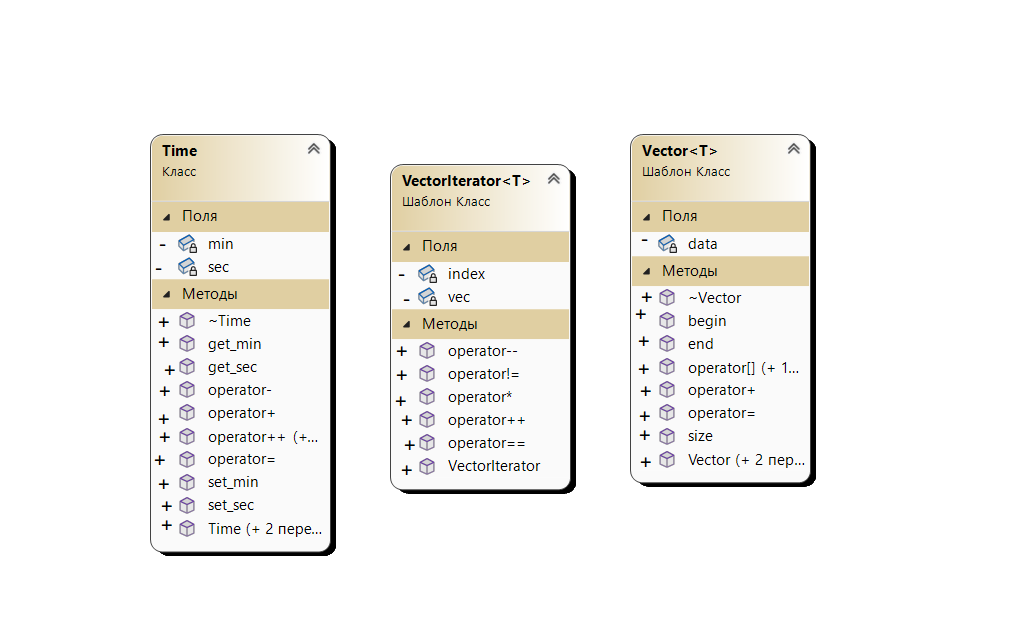
9. Инстанцировать шаблон для пользовательского класса.

10. Написать тестирующую программу, иллюстрирующую выполнение операций для контейнера, содержащего элементы пользовательского класса

Вариант 3



Описание класса



# Код программы(vector.h)

#include <vector>

#include <iostream>

template<typename T>

class VectorIterator;

template<typename T>

class Vector {

public:

Vector();

explicit Vector(int size) : data(size) {}

Vector(const Vector& other) : data(other.data) {}

Vector& operator=(const Vector& other) {

data = other.data;

return \*this;

}

~Vector() {};

T& operator[](int index) {

return data[index];

}

const T& operator[](int index) const {

return data[index];

}

friend Vector operator + (const Vector& a, const Vector& b) {

Vector result(a.size());

for (int i = 0; i < a.size(); ++i) {

result[i] = a[i] + b[i];

}

return result;

}

friend Vector operator + (const Vector& v, T x) {

Vector result(v.size());

for (int i = 0; i < v.size(); ++i) {

result[i] = v[i] + x;

}

return result;

}

Vector operator + (const Vector& other) const {

if (size != other.size) {

throw std::invalid\_argument("Vectors must be of equal size");

}

Vector<T> result(size);

for (int i = 0; i < size; i++) {

result[i] = data[i] + other.data[i];

}

return result;

}

VectorIterator<T> begin();

VectorIterator<T> end();

int size() const {

return data.size();

}

private:

std::vector<T> data;

};

template<typename T>

class VectorIterator {

public:

explicit VectorIterator(std::vector<T>& vec, int index) : vec(vec), index(index) {}

VectorIterator& operator++() {

++index;

return \*this;

}

VectorIterator& operator--() {

--index;

return \*this;

}

bool operator==(const VectorIterator& other) const {

return index == other.index;

}

bool operator!=(const VectorIterator& other) const {

return !(\*this == other);

}

T& operator\*() const {

return vec[index];

}

private:

std::vector<T>& vec;

int index;

};

template<typename T>

VectorIterator<T> Vector<T>::begin() {

return VectorIterator<T>(data, 0);

}

template<typename T>

VectorIterator<T> Vector<T>::end() {

return VectorIterator<T>(data, data.size());

}

template<typename T>

std::ostream& operator<<(std::ostream& os, const Vector<T>& v) {

os << "[";

for (int i = 0; i < v.size(); ++i) {

os << v[i];

if (i != v.size() - 1) {

os << ", ";

}

}

os << "]";

return os;

# }

# Код программы(Time.h)

#include <iostream>

using namespace std;

class Time

{

int min, sec;

public:

Time()

{

min = 0;

sec = 0;

};

Time(int m, int s)

{

min = m;

sec = s;

}

Time(const Time& t)

{

min = t.min;

sec = t.sec;

}

~Time() {};

int get\_min() { return min; }

int get\_sec() { return sec; }

void set\_min(int m) { min = m; }

void set\_sec(int s) { sec = s; }

// ïåðåãðóæåííûå îïåðàöèè

Time& operator=(const Time&);

Time& operator++();

Time operator++(int); // ïîñòôèêñíàÿ îïåðàöèÿ

Time operator+(const Time& other) const {

int totalsec = sec + other.sec;

int addmin = totalsec / 60;

totalsec %= 60;

int totalmin = min + other.min + addmin;

return Time(totalmin, totalsec);

}

Time operator-(const Time&);

// ãëîáàëüíûå ôóíêöèè ââîäà-âûâîäà

friend istream& operator>>(istream& in, Time& t);

friend ostream& operator<<(ostream& out, const Time& t);

};

Time& Time::operator=(const Time& t)

{

// ïðîâåðêà íà ñàìîïðèñâàèâàíèå

if (&t == this)

return \*this;

min = t.min;

sec = t.sec;

return \*this;

}

// ïåðåãðóçêà ïðåôèêñíîé îïåðàöèè èíêðåìåíò

Time& Time::operator++()

{

int temp = min \* 60 + sec;

temp++;

min = temp / 60;

sec = temp % 60;

return \*this;

}

// ïåðåãðóçêà ïîñòôèêñíîé îïåðàöèè èíêðåìåíò

Time Time::operator++(int)

{

int temp = min \* 60 + sec;

temp++;

Time t(min, sec);

min = temp / 60;

sec = temp % 60;

return t;

}

// ïåðåãðóçêà áèíàðíîé îïåðàöèè ñëîæåíèÿ

Time Time::operator-(const Time& t)

{

int temp1 = min \* 60 + sec;

int temp2 = t.min \* 60 + t.sec;

Time p;

p.min = (temp1 - temp2) / 60;

p.sec = (temp1 - temp2) % 60;

return p;

}

// ïåðåãðóçêà ãëîáàëüíîé ôóíêöèè-îïåðàöèè ââîäà

istream& operator>>(istream& in, Time& t)

{

cout << "min?";

in >> t.min;

cout << "sec?";

in >> t.sec;

return in;

}

// ïåðåãðóçêà ãëîáàëüíîé ôóíêöèè-îïåðàöèè âûâîäà

ostream& operator<<(ostream& out, const Time& t)

{

return (out << t.min << " : " << t.sec);

}

# Код программы(main)

#include "Vector.h"

#include "time.h"

#include <iostream>

#include <random>

int main() {

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_int\_distribution<> dis(0, 23);

std::uniform\_int\_distribution<> raz(0, 59);

Vector<int> v1(5);

Vector<double> v2(5);

for (int i = 0; i < v1.size(); ++i) {

v1[i] = i + 1;

v2[i] = (i + 1) \* 1.5;

}

Vector<Time> v5(5);

for (int i = 0; i < v5.size(); ++i) {

v5[i].set\_min(dis(gen));

v5[i].set\_sec(raz(gen));

}

Vector<Time> v6(5);

for (int i = 0; i < v6.size(); ++i) {

v6[i].set\_min(dis(gen));

v6[i].set\_sec(raz(gen));

}

std::cout << "v5 = " << v5 << std::endl;

std::cout << "v6 = " << v6 << std::endl;

std::cout << "v1 = " << v1 << std::endl;

std::cout << "v2 = " << v2 << std::endl;

Vector<Time> v3 = v5 + v6;

std::cout << "v3 = " << v3 << std::endl;

Vector<int> v7 = v1 + v1;

std::cout << "v7 = " << v7 << std::endl;

Time b;

b.set\_min(dis(gen));

b.set\_sec(raz(gen));

std::cout << b << std::endl;

Vector<Time> v4 = v3 + b;

std::cout << "v4 = " << v4 << std::endl;

Vector<int> v8 = v7 + 10;

std::cout << "v8 = " << v8 << std::endl;

int a;

VectorIterator<int> it = v1.begin();

a = \*it;

std::cout << "v1 before increment: " << a << std::endl;

++it;

a = \*it;

std::cout << "v1 after increment: " << a << std::endl;

--it;

a = \*it;

std::cout << "v1 after decrement: " << a << std::endl;

system("pause");

return 0;

}

# Работы программы

