Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.4**

Дисциплина: «информатика»

Тема: простое наследование

Вариант 5

Выполнил работу

студент группы РИС-20-1б

Зверев А.Д.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

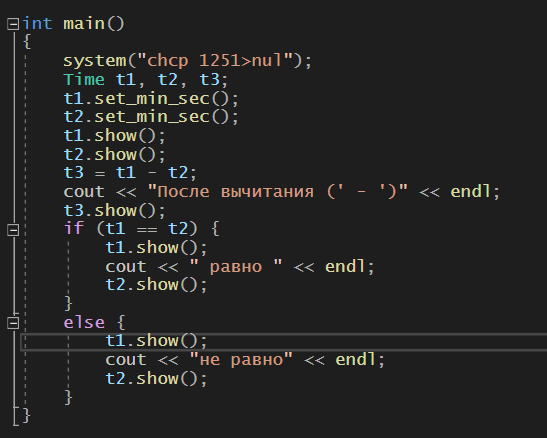
Пермь, 2021

**Постановка задачи**

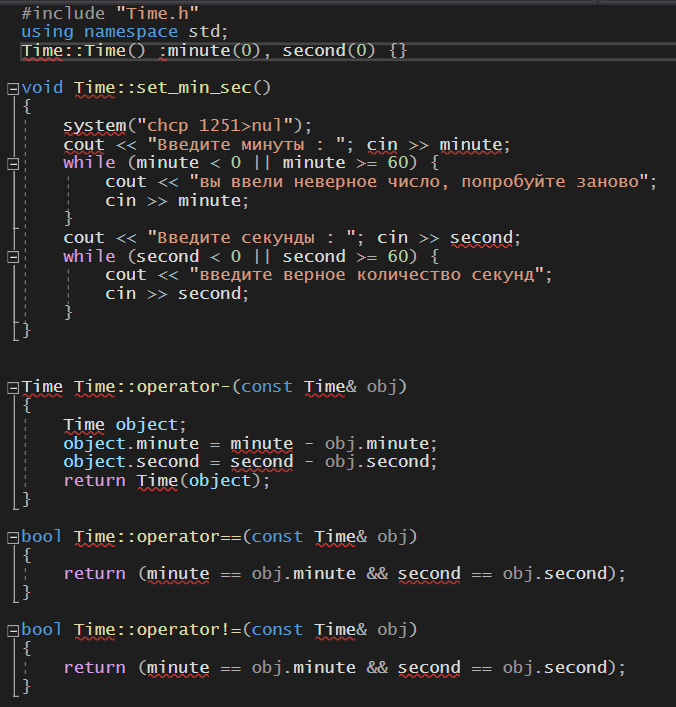
1. Определить пользовательский класс;
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
3. Определить в классе деструктор.
4. Определить в классе компоненты—функции для просмотра и установки полей данных (селекторы и модификаторы)
5. Перегрузить операцию присваивания
6. Перегрузить операции ввода и вывода объектов с помощью потоков.
7. Определить производный класс.
8. Написать программу, в которой продемонстрировать создание объектов и работу всех перегруженных операций.
9. Реализовать функции, получающие и возвращающие объект базового класса. Продемонстрировать принцип подстановки.
10. Создать базовый класс с парой чисел типа int.
11. Определить методы изменения полей и операцию сложения пар.
12. Создать производный класс с полями рубли и копейки.
13. Переопределить операции сложения и определить операции вычитания и деления базовых сумм.

**Анализ задачи**

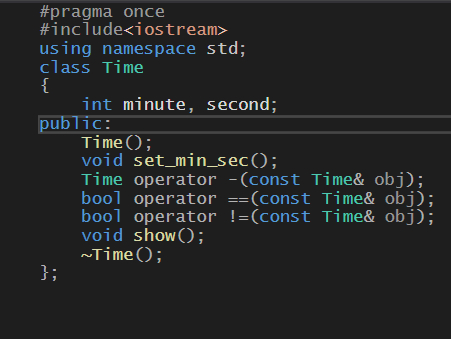
1. Для решения задачи необходимо…
   1. Написать функцию int main, в которой будет производиться вызов различных функций и получение параметров;



* 1. В отдельном файле описать ввод и вывод элементов, а также различные внутренние проверки;



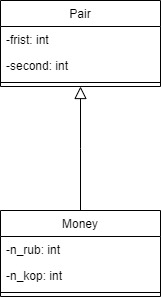
* 1. В заголовочном файле Time.h описать класс с нужными переменными;



1. В программе были использованы следующие типы данных:
   1. Тип данных int для хранения введенного времени;



**Блок—схема программы**



**Решение**

#include"Time.h"

using namespace std;

int main()

{

system("chcp 1251>nul");

Time t1, t2, t3;

t1.set\_min\_sec();

t2.set\_min\_sec();

t1.show();

t2.show();

t3 = t1 - t2;

cout << "После вычитания (' - ')" << endl;

t3.show();

if (t1 == t2) {

t1.show();

cout << " равно " << endl;

t2.show();

}

else {

t1.show();

cout << "не равно" << endl;

t2.show();

}

}

#include "Time.h"

using namespace std;

Time::Time() :minute(0), second(0) {}

void Time::set\_min\_sec()

{

system("chcp 1251>nul");

cout << "Введите минуты : "; cin >> minute;

while (minute < 0 || minute >= 60) {

cout << "вы ввели неверное число, попробуйте заново";

cin >> minute;

}

cout << "Введите секунды : "; cin >> second;

while (second < 0 || second >= 60) {

cout << "введите верное количество секунд";

cin >> second;

}

}

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

{

Time object;

object.minute = minute - obj.minute;

object.second = second - obj.second;

return Time(object);

}

bool Time::operator==(const Time& obj)

{

return (minute == obj.minute && second == obj.second);

}

bool Time::operator!=(const Time& obj)

{

return (minute == obj.minute && second == obj.second);

}

void Time::show()

{

if (minute < 0) {

cout << "вы повели себя не хорошо, перезапустите программу";

}

cout << minute << ":" << second << endl;

}

Time::~Time() {}

#pragma once

#include<iostream>

using namespace std;

class Time

{

int minute, second;

public:

Time();

void set\_min\_sec();

Time operator -(const Time& obj);

bool operator ==(const Time& obj);

bool operator !=(const Time& obj);

void show();

~Time();

};

**Скриншоты**

