

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.01 – «Информатика и вычислительная
техника»

**Отчет по
лабораторной работе 5 (классы)**

Выполнил студент гр. ИВТ-23-16
Давыдов Андрей Юрьевич

Проверил:

ст. преп. каф. ИТАС

Яруллин Денис Владимирович

(оценка)

(подпись)

(дата)

г. Пермь, 2023

Постановка задачи

Базовый класс:

ТРОЙКА_ЧИСЕЛ (TRIAD)

Первое_число (first) - int

Второе_число (second) – int

Третье_число (third) - int

Определить методы изменения полей и сравнения триады.

Создать производный класс TIME с полями часы, минуты и секунды. Определить полный набор операций сравнения временных промежутков.

Код программы:

```
Object.h  Vector.h  Time.h  Triad.h  cl_5.cpp
cl_5
1  #pragma once
2  #include <iostream>
3
4  using namespace std;
5
6  // Абстрактный базовый класс Object
7  class Object {
8  public:
9      virtual void Show() const = 0; // Чисто виртуальная функция
10     virtual ~Object() {}
11 };
12
```

```
Object.h  Vector.h  Time.h  Triad.h  cl_5.cpp
cl_5
1  #pragma once
2  #include "Object.h"
3
4  class Triad : public Object {
5  protected:
6      int first;
7      int second;
8      int third;
9
10  public:
11      Triad(int f = 0, int s = 0, int t = 0) : first(f), second(s), third(t) {}
12      virtual void Show() const override {
13          cout << "(" << first << ", " << second << ", " << third << ")" << endl;
14      }
15      virtual ~Triad() {}
16
17      virtual void setFirst(int f) { first = f; }
18      virtual void setSecond(int s) { second = s; }
19      virtual void setThird(int t) { third = t; }
20
21      virtual int getFirst() const { return first; }
22      virtual int getSecond() const { return second; }
23      virtual int getThird() const { return third; }
24
25      virtual bool isEqual(const Triad& other) const = 0; // Чисто виртуальная функция
26
27      friend ostream& operator<<(ostream& out, const Triad& triad) {
28          out << "(" << triad.getFirst() << ", " << triad.getSecond() << ", " << triad.getThird() << ")";
29          return out;
30      }
31 };
32
33
```

```
Object.h  Vector.h  Time.h  Triad.h  cl_5.cpp
cl_5
1  #pragma once
2  #include "Triad.h"
3
4  class Time : public Triad {
5  public:
6      Time(int h = 0, int m = 0, int s = 0) : Triad(h, m, s) {}
7      virtual void Show() const override {
8          cout << "Time: " << first << "h " << second << "m " << third << "s" << endl;
9      }
10     void setHours(int h) { setFirst(h); }
11     void setMinutes(int m) { setSecond(m); }
12     void setSeconds(int s) { setThird(s); }
13
14     int getHours() const { return getFirst(); }
15     int getMinutes() const { return getSecond(); }
16     int getSeconds() const { return getThird(); }
17
18     bool isEqual(const Triad& other) const override {
19         const Time& o = static_cast<const Time&>(other);
20         return getHours() == o.getHours() &&
21             getMinutes() == o.getMinutes() &&
22             getSeconds() == o.getSeconds();
23     }
24
25     // Определение операций сравнения для Time
26     bool operator==(const Time& other) const {
27         return isEqual(other);
28     }
29
30     bool operator!=(const Time& other) const {
31         return !isEqual(other);
32     }
33
34     bool operator<(const Time& other) const {
35         // Сравниваем по общему количеству секунд
36         return getTotalSeconds() < other.getTotalSeconds();
37     }
38
39     bool operator>(const Time& other) const {
40         return getTotalSeconds() > other.getTotalSeconds();
41     }
42
43     bool operator<=(const Time& other) const {
44         return getTotalSeconds() <= other.getTotalSeconds();
45     }
46
47     bool operator>=(const Time& other) const {
48         return getTotalSeconds() >= other.getTotalSeconds();
49     }
50
51 private:
52     int getTotalSeconds() const {
53         return getHours() * 3600 + getMinutes() * 60 + getSeconds();
54     }
55 };
56
```

```
Object.h  Vector.h  Time.h  Triad.h  cl_5.cpp
cl_5  Vector

1  #pragma once
2  #include <vector>
3  #include "Object.h"
4
5  class Vector {
6  private:
7      vector<Object*> items;
8
9  public:
10     Vector() {}
11     ~Vector() {
12         for (auto item : items) {
13             delete item;
14         }
15     }
16
17     void Add(Object* item) {
18         items.push_back(item);
19     }
20
21     friend ostream& operator<<(ostream& out, const Vector& v) {
22         if (v.items.empty()) {
23             out << "Пусто :(" << endl;
24         }
25         else {
26             for (const auto& item : v.items) {
27                 item->Show();
28             }
29         }
30         return out;
31     }
32 };
```

```
Object.h  Vector.h  Time.h  Triad.h  cl_5.cpp  (Глобальн
cl_5

1  #include "Vector.h"
2  #include "Time.h"
3  #include "Triad.h"
4
5  int main() {
6      Vector vector;
7
8      // Создание объектов Triad и Time
9      Object* t1 = new Time(1, 2, 3);
10     Object* t2 = new Time(12, 30, 45);
11
12     // Добавление объектов в вектор
13     vector.Add(t1);
14     vector.Add(t2);
15
16     // Вывод содержимого вектора
17     cout << vector;
18
19
20
21     return 0;
22 }
23
```