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**

**по лабораторной работе № 18.2**

Дисциплина: основы алгоритмизации и программирования

Тема: «Классы и объекты. Использование конструкторов»

Вариант 1

Выполнил работу:

студент группы РИС-20-1Б

Азмагулов Артём Вадимович

Проверила

Доцент кафедры ИТАС

Полякова О.А.

Пермь

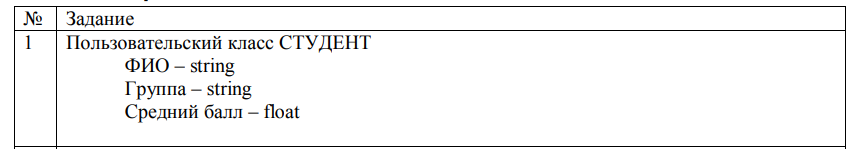
2021

**Цель работы**

1. Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio
2. Создание объектов с использованием конструкторов

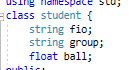
**Постановка задачи**

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. Написать демонстрационную программу, в которой продемонстрировать все три случая вызова конструктора – копирования, вызов конструктора с параметрами и конструктора без параметров.

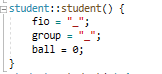


**Анализ задачи**

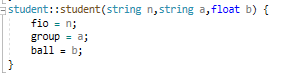
1. Необходимые действия
2. Объявить в классе Student скрытые поля для хранения ФИО, должности и зарплаты сотрудника



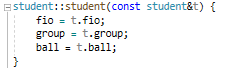
1. Реализовать конструктор без параметров, который будет обнулять значение рейтинга , присваивать пустую строку ФИО и группу студента



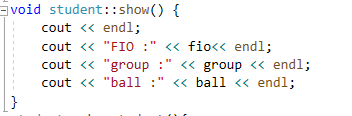
1. Реализовать конструктор с параметрами, который будет принимать значения ФИО, рейтинг студента и присваивать их соответствующим полям класса



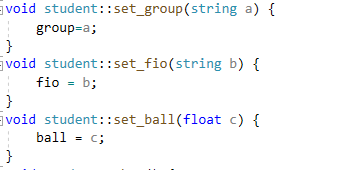
1. Реализовать конструктор копирования, который будет принимать константную ссылку на объект класса



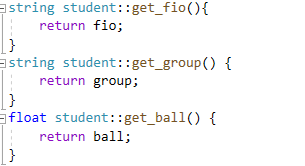
1. Реализовать функцию вывода информации о сотруднике



1. Реализовать модификаторы:



1. Реализовать селекторы, которые будут выводить значения полей класса по отдельности



Основная программа

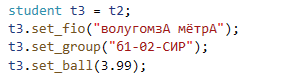
1. Создать объект класса, используя конструктор без параметров, и вывести информацию о нем



1. Создать объект класса, используя конструктор с параметрами, и вывести информацию о нем



1. Создать объект класса и присвоить значения полям объекта, используя конструктор копирования. Вывести информацию об объекте



1. Программный код

18.сpp

#include <iostream>

using namespace std;

class student {

string fio;

string group;

float ball;

public:

student();

student(string, string, float);

student(const student&);

~student();

string get\_fio();

void set\_fio(string);

string get\_group();

void set\_group(string);

float get\_ball();

void set\_ball(float);

void show();

};

student::student() {

fio = "\_";

group = "\_";

ball = 0;

}

student::student(string n,string a,float b) {

fio = n;

group = a;

ball = b;

}

student::student(const student&t) {

fio = t.fio;

group = t.group;

ball = t.ball;

}

student::~student() {

cout << "Destructor" << this<<endl;

}

string student::get\_fio(){

return fio;

}

string student::get\_group() {

return group;

}

float student::get\_ball() {

return ball;

}

void student::set\_group(string a) {

group=a;

}

void student::set\_fio(string b) {

fio = b;

}

void student::set\_ball(float c) {

ball = c;

}

void student::show() {

cout << endl;

cout << "FIO :" << fio<< endl;

cout << "group :" << group << endl;

cout << "ball :" << ball << endl;

}

student make\_student(){

string s,b;

float i;

cout << endl;

cout << "FIO: " << endl;

cin >> s;

cout << "group: " << endl;

cin >> b;

cout << "ball(myach): " << endl;

cin >> i;

student t(s, b, i);

return t;

}

void print\_student(student t) {

t.show();

}

void main() {

setlocale(LC\_ALL, "rus");

student t1;

t1.show();

student t2("Азмагулов Артём В.", "РИС-20-1б", 4.01);

t2.show();

student t3 = t2;

t3.set\_fio("волугомзА мётрА");

t3.set\_group("б1-02-СИР");

t3.set\_ball(3.99);

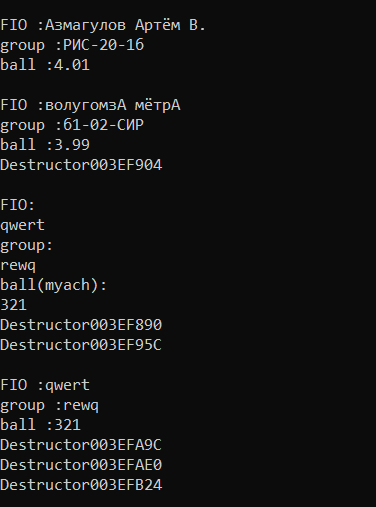
print\_student(t3);

t1 = make\_student();

t1.show();

}

1. Демонстрация работы программы



1. Контрольные вопросы
2. Для чего нужен конструктор?

- для инициализации полей объекта

1. Сколько типов конструкторов существует в С++?

- существует три типа: конструктор с параметрами, конструктор без параметров, конструктор копирования

1. Для чего используется деструктор? В каких случаях деструктор описывается явно?

- используется для высвобождения ресурсов, которые используются объектом. Его нужно указывать явно, когда происходит работа с динамической памятью

1. Для чего используется конструктор без параметров? Конструктор с параметрами? Конструктор копирования?

- конструктор без параметров используется создания «пустого» объекта

- конструктор с параметрами используется для инициализации объекта требуемыми значениями

- конструктор копирования используется для создания нового объекта путем копирования данных старого объекта

1. В каких случаях вызывается конструктор копирования?

- при описании нового объекта с инициализацией с другим объектом;

- при передаче объекта в функцию по значению;

- при возврате объекта из функции

1. Перечислить свойства конструкторов

- не возвращает значение, даже типа void. Нельзя получить указатель на конструктор;

- класс может иметь несколько конструкторов с разными параметрами для разных видов инициализации;

- конструктор без параметров называется конструктором по умолчанию;

- параметры конструктора могут иметь любой тип

- если программист не указал ни одного конструктора, компилятор создает его автоматически. Такой конструктор вызывает конструкторы по умолчанию. В случае, когда класс содержит константы или ссылки, при попытке создания объекта класса будет выданы ошибка, так как их необходимо инициализировать конкретными значениями, а конструктор по умолчанию этого делать не умеет

- конструкторы не наследуются

- конструкторы нельзя описывать с модификаторами const, virtual и static

- конструкторы глобальных объектов вызываются до функции main. Локальные объекты создаются, как только становится активной область их действия. Конструктор запускается и при создании временного объекта

1. Перечислить свойства деструкторов

- не имеет аргументов и возвращаемого значения

- не наследуется

- не может быть объявлен как const или static

- может быть виртуальным

1. К каким атрибутам имеют доступ методы класса?

- методы класса имеют доступ ко всем атрибутам данного класса

1. Что представляет собой указатель this?

- представляет собой неявно определенный указатель на сам объект

1. Какая разница между методами определенными внутри класса и вне класса?

- методы класса имеют доступ ко всем полям класса, в независимости от спецификаторов

1. Какое значение возвращает конструктор?

- конструктор не возвращает значение

1. Какие методы создаются по умолчанию?

- конструктор без параметров и конструктор копирования

1. Какое значение возвращает деструктор?

- деструктор не возвращает значение

1. Дано описание класса

class Student {

string name;

int group;

public:

student(string, int);

Student(const Student&);

~Student();

};

Какой метод отсутствует в описании класса?

- отсутствует конструктор без параметров

1. Какой метод будет вызван при выполнении следующих операторов

Student\* s;

s = new Student;

- конструктор без параметров

1. Какой метод будет вызван при выполнении следующих операторов

Student s(“Ivanov”, 20);

- конструктор с параметрами

1. Какие методы будут вызваны при выполнении следующих операторов

Student s1(“Ivanov”, 20);

Student s2 = s1;

- конструктор с параметрами и конструктор копирования

1. Какие методы будут вызваны при выполнении следующих операторов

Student s1(“Ivanov”, 20);

Student s2;

s2 = s1;

- конструктор без параметров, с параметрами, копирования

1. Какой конструктор будет использоваться при передаче параметра в функцию print()

Void print(student a){

a.show();

}

- конструктор копирования

1. Класс описан следующим образом

class Student{

string name;

int age;

public:

void set\_name(string);

void set\_age(int);

};

Student p;

Каким образом можно присвоить новое значение атрибуту name объекта p?

p.set\_name(“Vasiliy”);

**Блок-схема**

Описание полей класса

