Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа по ООП**

**«№4»**

Выполнил:

студент группы РИС-23-2б

Ившин Максим Сергеевич

Проверила:

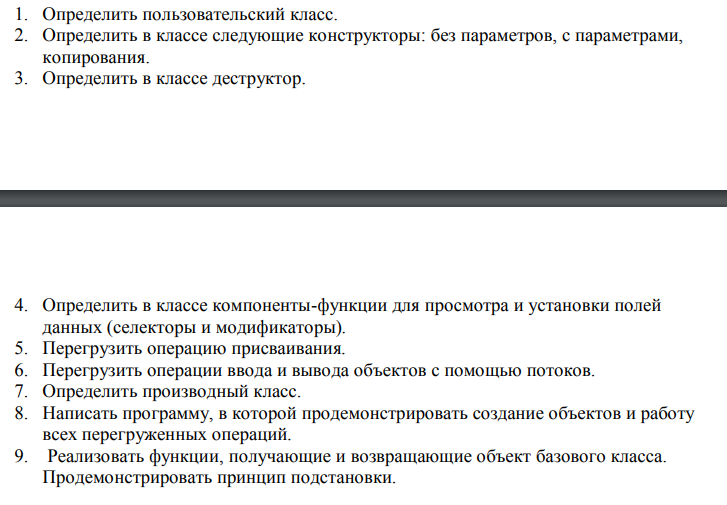
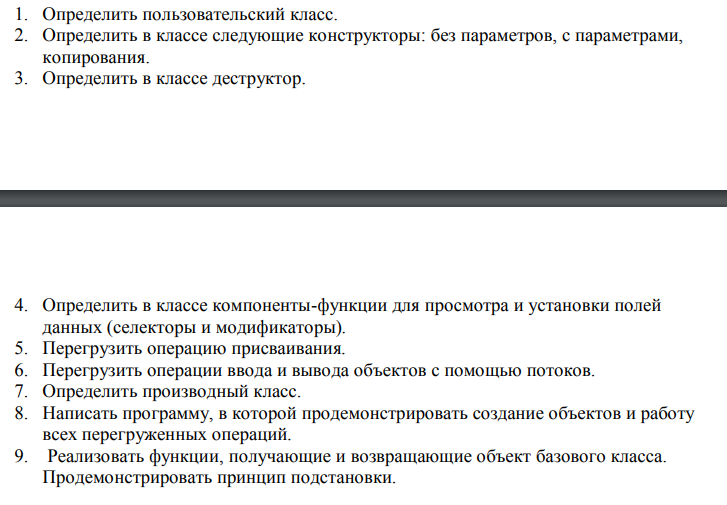
доцент кафедры ИТАС

О.А. Полякова

2024 г.

**Разработка алгоритма**

**Постановка задачи:**

** Анализ задачи:**

1. Будем использовать несколько файлов для удобной работы с кодом.
2. Создадим пользовательский класс Person и производный класс Student.
3. Добавим конструктор без параметров, с параметрами, конструктор копирования для обоих классов. Также определим пустые деструкторы.
4. Создадим геттеры и сеттеры в соответствии с поставленной задачей, перегрузим функцию ввода/вывода для обоих классов.
5. Напишем код для проверки реализации всех функций.

**Код программы на C++:**

main.cpp:

#include <iostream>

#include "Person.cpp"

using namespace std;

int main()

{

    // system("chcp 1251>null");

    Person person1("Нашульпанов Денис Васильевич", 35);

    cout << person1;

    Person person2;

    cout << person2;

    Person person3(person1);

    cout << person3;

    Student student1("Виталий Иваанович Наливкин", 20, 0);

    cout << student1;

    student1.setYoS(1);

    cout << student1;

    student1++;

    cout << student1;

    Student student2;

    cout << student2 << endl << endl;

    cin >> student2;

    Student student3;

    student3 = student2;

    cout << student3 << endl;

    return 0;

    cout << student3;

}

Person.cpp:

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Person

{

protected:

    string name;

    int age;

public:

    Person();

    Person(string, int);

    Person(const Person &);

    ~Person();

    void setName(const string);

    void setAge(const int);

    void operator=(const Person &);

    friend ostream &operator<<(ostream &, Person &);

    friend istream &operator>>(istream &, Person &);

};

class Student : public Person

{

protected:

    int year\_of\_study;

public:

    Student();

    Student(string, int, int);

    void setYoS(int);

    Student &operator++();

    Student &operator++(int);

    friend ostream &operator<<(ostream &, const Student &);

    friend istream &operator>>(istream &, Student &);

};

Student &Student::operator++()

{

    ++this->year\_of\_study;

    return \*this;

}

Student &Student::operator++(int)

{

    this->year\_of\_study++;

    return \*this;

}

Person::Person()

{

    cout << "Вызвался конструктор без параметров для объекта " << this << endl;

    this->name = "-";

    this->age = 0;

}

Person::Person(string name, int age)

{

    cout << "Вызвался конструктор с параметрами для объекта " << this << endl;

    this->name = name;

    this->age = age;

}

Person::Person(const Person &other)

{

    cout << "Вызвался конструктор копирования для объекта " << this << endl;

    this->name = other.name;

    this->age = other.age;

}

Person::~Person()

{

    cout << "Вызвался деструктор для объекта " << this << endl;

}

void Person::setName(const string name)

{

    this->name = name;

}

void Person::setAge(const int age)

{

    this->age = age;

}

void Person::operator=(const Person &person)

{

    this->name = person.name;

    this->age = person.age;

}

ostream &operator<<(ostream &stream, Person &person)

{

    stream << "ФИО: " << person.name << endl

           << "Возраст: " << person.age << "\n\n";

    return stream;

}

istream &operator>>(istream &stream, Person &person)

{

    string name;

    int age;

    cout << "Введите имя: ";

    cin.seekg(cin.eof());

    getline(cin, name);

    cout << "Введите возраст: ";

    cin >> age;

    person.setName(name);

    person.setAge(age);

    return stream;

}

Student::Student()

{

    this->name = "-";

    this->age = 0;

    this->year\_of\_study = 0;

}

Student::Student(string name, int age, int yos)

{

    this->name = name;

    this->age = age;

    this->year\_of\_study = yos;

}

void Student::setYoS(int yos)

{

    this->year\_of\_study = yos;

}

ostream &operator<<(ostream &stream, const Student &student)

{

    stream << "Имя: " << student.name << endl

           << "Возраст: " << student.age << endl

           << "Год обучения: " << student.year\_of\_study << endl << endl;

    return stream;

}

istream &operator>>(istream &stream, Student &student)

{

    string name;

    int yos, age;

    cout << "Введите ФИО: ";

    cin.seekg(cin.eof());

    getline(stream, student.name);

    cout << "Введите возраст: ";

    stream >> age;

    student.age = age;

    cout << "Введите год обучения: ";

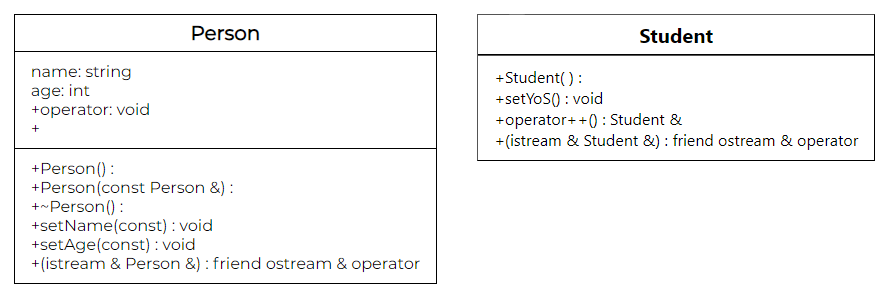
    stream >> yos;

    student.setYoS(yos);

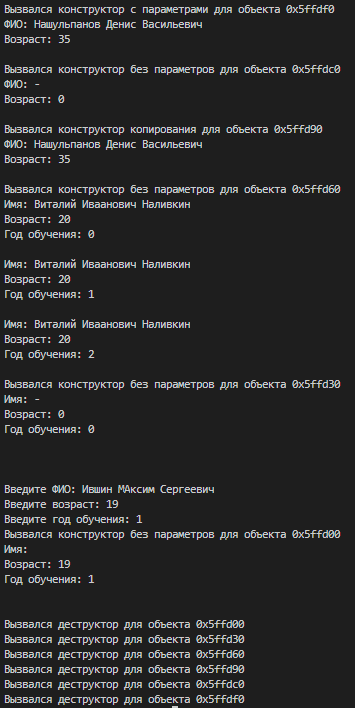
    return stream;

}

Диаграмма класса:



Пример работы программы:



**GitHub:** *https://github.com/geroineee/PNRPU*