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.Программа управляемая событиями ."

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 иерархию пользовательских классов (см. лабораторную работу №5). Во главе иерархии должен стоять абстрактный класс с чисто виртуальными методами для ввода и вывода информации об атрибутах объектов.

2. Реализовать конструкторы, деструктор, операцию присваивания, селекторы и модификаторы.

3. Определить класс-группу на основе структуры, указанной в варианте.

4. Для группы реализовать конструкторы, деструктор, методы для добавления и удаления элементов в группу, метод для просмотра группы, перегрузить операцию для получения информации о размере группы.

5. Определить класс Диалог – наследника группы, в котором реализовать методы для обработки событий.

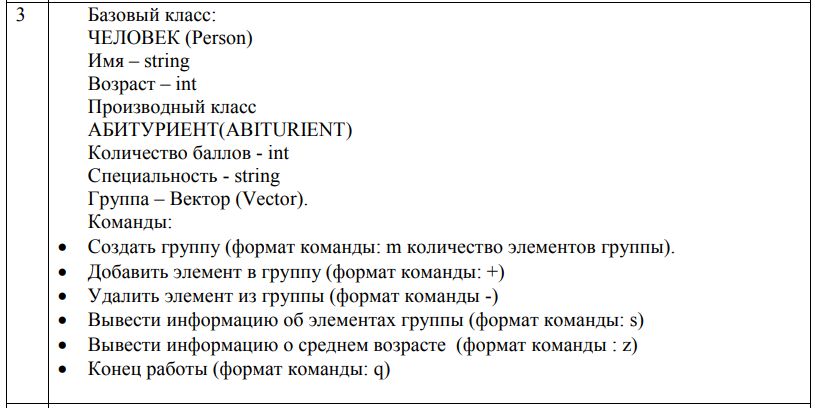
6. Добавить методы для обработки событий группой и объектами пользовательских классов.

7. Написать тестирующую программу.

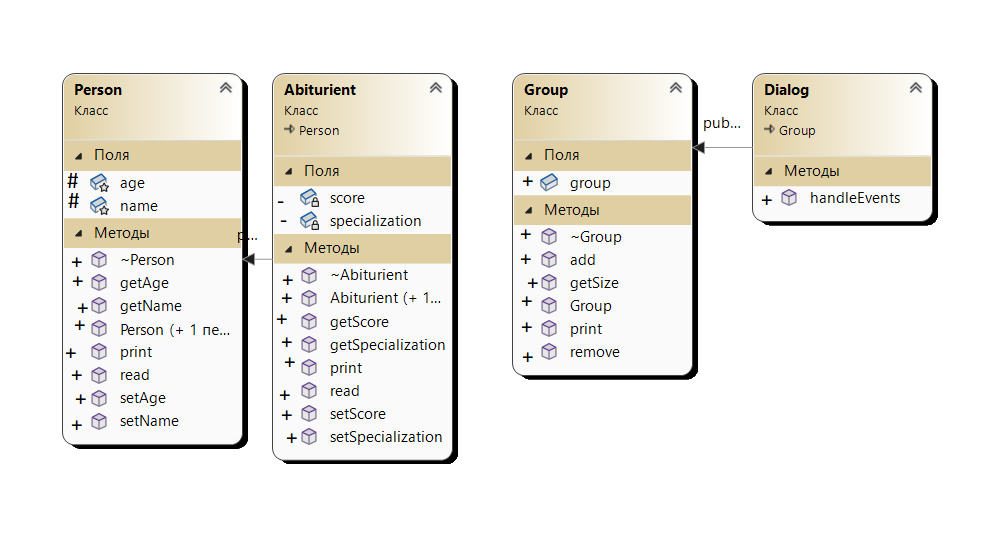
8. Нарисовать тестирующую программу.

8. Нарисовать диаграмму классов и диаграмму объектов.

Вариант 3



Описание класса



# Код программы(group.h)

#pragma once

#include <vector>

class Group {

public:

vector<Person\*> group;

Group() {};

~Group() {

for (auto& p : group) {

delete p;

}

};

void add(Person\* p) {

group.push\_back(p);

};

void remove(int index) {

if (index >= 0 && index < group.size()) {

delete group[index];

group.erase(group.begin() + index);

}

};

void print() const {

for (auto& p : group) {

p->print();

}

};

int getSize() const {

return group.size();

};

};

# Код программы(abiturient.h)

#include "person.h"

#include <string>

class Abiturient : public Person {

private:

int score;

string specialization;

public:

Abiturient() {};

Abiturient(const string& name\_, int age\_, int score\_, const string& specialization\_)

: Person(name\_, age\_), score(score\_), specialization(specialization\_) {};

~Abiturient() {};

void print() const override {

cout << "Name: " << name << ", Age: " << age << ", Score: " << score << ", Specialization: " << specialization << endl;

};

void read() override {

Person::read();

cout << "Score: ";

cin >> score;

cout << "Specialization: ";

cin >> specialization;

};

int getScore() const {

return score;

};

void setScore(int score\_) {

score = score\_;

};

const string& getSpecialization() const {

return specialization;

};

void setSpecialization(const string& specialization\_) {

specialization = specialization\_;

};

};

# Код программы(dialog.h)

#include "group.h"

class Dialog : public Group {

public:

void handleEvents() {

char command = ' ';

while (command != 'q') {

cout << "Enter command (+/-/s/z/q): ";

cin >> command;

switch (command) {

case '+': {

Abiturient\* a = new Abiturient();

a->read();

add(a);

break;

}

case '-': {

int index;

cout << "Enter index: ";

cin >> index;

remove(index);

break;

}

case 's': {

print();

break;

}

case 'z': {

int sum = 0;

for (auto& p : group) {

sum += p->getAge();

}

if (group.size() > 0) {

cout << "Average age: " << (sum / group.size()) << endl;

}

break;

}

}

}

};

};

# Код программы(person.h)

#pragma once

using namespace std;

class Person {

protected:

string name;

int age;

public:

Person() {};

Person(const string& name\_, int age\_) : name(name\_), age(age\_) {};

virtual ~Person() {};

virtual void print() const = 0;

virtual void read() {

cout << "Name: ";

cin >> name;

cout << "Age: ";

cin >> age;

};

const string& getName() const {

return name;

};

void setName(const string& name\_) {

name = name\_;

};

int getAge() const {

return age;

};

void setAge(int age\_) {

age = age\_;

};

};

# Код программы(main)

#include <iostream>

#include "abiturient.h"

#include "dialog.h"

#include "group.h"

#include "person.h"

int main() {

Dialog d;

d.handleEvents();

return 0;

}

# Работы программы

