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**

Дисциплина: Основы алгоритмизации и программирования

Тема: Сохранение данных в файле с использованием потоков

Выполнил работу

студент группы РИС-22-1б

Рыжков Н.С.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2023

**ТЕХНИЧЕСКАЯ СПЕЦИФИКАЦИЯ ДЛЯ ЛАБОРАТОРНОЙ РАБОТЫ №8**

Базовый класс: ЧЕЛОВЕК (Person)

Имя – string, Возраст – int

Производный класс

АБИТУРИЕНТ(ABITURIENT)

Количество баллов – int, Специальность – string, Группа – Вектор (Vector).

Команды:

* Создать группу (формат команды: m количество элементов группы).
* Добавить элемент в группу (формат команды: +)
* Удалить элемент из группы (формат команды -)
* Вывести информацию об элементах группы (формат команды: s)
* Вывести информацию о среднем возрасте (формат команды : z)
* Конец работы (формат команды: q)

Класс Person представляет собой абстрактный класс, который содержит информацию о человеке, такую как имя и возраст. Он также содержит виртуальные методы для чтения и вывода информации о человеке.

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\_;

};

};

Класс Group представляет собой группу людей, которые могут быть экземплярами класса Person или его производных классов. Он содержит вектор указателей на объекты класса Person, а также методы для добавления, удаления и вывода информации о людях в группе.

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(); };

Класс Abiturient является производным классом от класса Person и содержит дополнительную информацию о студентах, такую как баллы и специализация. Он также содержит методы для чтения и вывода информации о студентах.

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\_;

};

};

Класс Dialog является производным классом от класса Group и представляет собой диалоговое окно, которое позволяет пользователю добавлять, удалять и просматривать информацию о студентах. Он содержит метод handleEvents(), который обрабатывает события, связанные с добавлением, удалением и выводом информации о студентах.

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;

}

}

}

};

};

Главная функция создает экземпляр класса Dialog и вызывает его метод handleEvents().

int main() {

Dialog d;

d.handleEvents();

return 0;

}

**Блок-схема программы**

