

Написать программу, позволяющую формировать список студентов академической группы. Интерфейс программы должен предоставлять пользователю следующие возможности:

- a. добавление студентов в группу;
- b. удаление студента из группы;
- c. модификация данных студента;
- d. печать академической группы;
- e. сортировка списка студентов;
- f. поиск студента по заданному критерию;
- g. сохранение списка студентов в файл (при завершении работы программы);
- h. загрузка списка студентов из файла (при старте программы);
- i. выход из программы.

При разработке программы следует использовать двумерный динамический массив для хранения списка студентов (одномерный динамический массив указателей на объекты класса Student).

Для реализации функциональности программы необходимо разработать следующие классы:

```
class Person
{
protected:
    char *name;
    char *surname;
    int age;
public:
    Person();
    Person(const char * name, const char * surname, int age);
    Person(const Person &obj);
```

```

    Person(Person &&obj);
    ~Person();
    const char* getName()const;
    const char * getSurname()const;
    int getAge() const;
    void setName(const char * name);
    void setSurname(const char * surname);
    void setAge(int age);
    void Print()const;
    Person& operator=(const Person &obj);
    Person& operator=(Person &&obj);
    friend std::istream& operator>>(std::istream &is, Person& obj);
    friend std::ostream& operator<<(std::ostream &os, const Person& obj);
};

class Student : public Person
{
    char phone[14];
    double rating;
public:
    Student();
    Student(const char * name, const char * surname, int age, const char * phone,
            double rating);
    Student(const Student &obj);
    Student(Student &&obj);
    double getRating() const;
    const char * getPhone() const;
    void setPhone(const char * phone);
    void setRating(double average);
    void Print() const;
    Student& operator=(const Student &obj);
    Student& operator=(Student &&obj);
    friend std::istream& operator>>(std::istream &is, Student& obj);
    friend std::ostream& operator<<(std::ostream &os, const Student& obj);
};

class AcademyGroup
{
    Student ** pSt;
    int count;

```

```
public:
    AcademyGroup();
    AcademyGroup(const AcademyGroup& obj);
    AcademyGroup(AcademyGroup &&obj);
    ~AcademyGroup();
    void AddStudents();
    void DeleteStudent();
    void EditStudent();
    void Print();
    void FindStudent();
    void Sort();
    void Save();
    void Load();
    AcademyGroup& operator=(const AcademyGroup &obj);
    AcademyGroup& operator=(AcademyGroup &&obj);
    Student& operator[](int index);
    friend std::ostream& operator<<(std::ostream &os, AcademyGroup& obj);
};
```