



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)**

Федеральное государственное бюджетное образовательное учреждение высшего образования

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)


О т ч е т

по домашнему заданию № 4

Название домашней работы: Основы объектно-ориентированного
программирования.

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-15Б


(Подпись, дата)

В.А. Бирюков
(И.О. Фамилия)

Преподаватель


(Подпись, дата)

А.А. Веселовский
(И.О. Фамилия)

Москва, 2024

Вариант 2

Задание: Разработать класс для реализации указанного объекта. Построить его диаграмму и составить программный код.

Объект – ученик. Поля: фамилия, год рождения, средняя годовая оценка по пятибалльной шкале. Методы: процедура инициализации, процедура вывода информации об объекте на экран, функция вычисления возраста ученика и функции, возвращающие значения полей по запросу.

Напишем код класса на языке C++

```
class Student{
private:
    std::string Surname;
    int BirthYear;
    float Mark;
public:
    Student(std::string S, int Y, float M): Surname(S), BirthYear(Y),
Mark(M) {}
    Student(): Surname(""),BirthYear(0),Mark(0) {}
    void ShowInfo(){
        std::cout << "Ученик " << Surname << " родился в " << BirthYear
        << " году. Средняя годовая оценка: " << Mark;
    }
    int Age(int ThisYear){
        return ThisYear - BirthYear;
    }
    int getBirthYear(){
        return BirthYear;
    }
    int getMark(){
        return Mark;
    }
    std::string getSurname(){
        return Surname;
    }
};
```

Составим диаграмму класса:

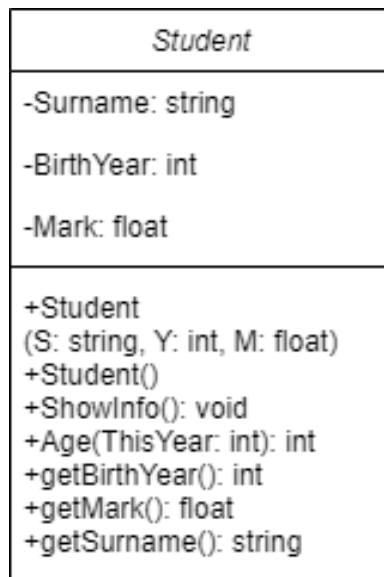


Рисунок 1 – диаграмма класса Student

Часть 1. Массив объектов

Задание: разработать тестирующую программу, в которой создается группа из *n* учеников, для которой определяется процент совершеннолетних и процент отличников.

Напишем код программы на C++

```
int main(){
    int Year = 2024;
    int Otl_count = 0, Sov_count = 0;
    int const num = 3;
    float PercOtl, PercSov;
    Student Group[num];

    Group[0] = Student("Земсков", 2007, 3.2);
    Group[1] = Student("Иванов", 2004, 2.4);
    Group[2] = Student("Петров", 1998, 5.0);

    for (int i = 0; i < num; i++){
        Group[i].ShowInfo();
        std::cout << std::endl;
        if (Group[i].Age(Year) >= 18) {
            Sov_count++;
        }
    }
}
```

```

        if (Group[i].getMark() == 5.0) {
            Otl_count++;
        }
    }

    PercSov = (static_cast<float>(Sov_count)/num*100);
    PercOtl = (static_cast<float>(Otl_count)/num*100);

    std::cout << "Процент совершеннолетних в группе: " << PercSov <<
std::endl;
    std::cout << "Процент отличников в группе: " << PercOtl << std::endl;

    return 0;
}

```

Запустим программу и посмотрим на вывод

```

[Running] cd "d:\Лабы\АиП\Dz_4\" && g++ main.cpp -o main && "d:\Лабы\АиП\Dz_4\"main
Ученик Земсков родился в 2007 году. Средняя годовая оценка: 3.2
Ученик Иванов родился в 2004 году. Средняя годовая оценка: 2.4
Ученик Петров родился в 1998 году. Средняя годовая оценка: 5
Процент совершеннолетних в группе: 66.6667
Процент отличников в группе: 33.3333

[Done] exited with code=0 in 0.544 seconds

```

Рисунок 2 – результат работы программы

Программа работает корректно на тестовых данных.

Часть 2. Композиция

Задание: используя разработанный в первой части задания класс, построить диаграмму классов композиционного объекта. Составить программный код описания класса. Разработать тестирующую программу.

Объект – учебный класс. Включает в себя количество учеников в классе и массив учеников. Методы объекта должны позволять: инициализировать класс конкретными учениками, вывести информацию обо всех учениках на экран, получить количество совершеннолетних учеников в классе, определить процент отличников.

Напишем код класса и тестирующей программы на C++

```
#include <iostream>
#include <vector>

const int curr_year = 2024;

class Student {
private:
    std::string surname;
    int birthYear;
    float mark;

public:
    Student(std::string surname, int birthYear, float mark):
        surname(surname), birthYear(birthYear), mark(mark) {}

    void showInfo() {
        std::cout << "Ученик " << surname << " родился в " << birthYear
            << " году. Средняя годовая оценка: " << mark << std::endl;
    }

    int Age() {
        return curr_year - birthYear;
    }

    float getMark() {
        return mark;
    }

    std::string getSurname() {
        return surname;
    }

    float getBirthYear(){
        return birthYear;
    }
};

class Group {
private:
    std::vector<Student> students;

public:
    void addStudent(std::string surname, int birthYear, float mark) {
        students.emplace_back(surname, birthYear, mark);
    }

    void showGroupInfo(){
        std::cout << "Информация обо всех учениках группы: " << std::endl;
        for (auto& student : students) {
            student.showInfo();
        }
    }
};
```

```

    }

    int countAdults(){
        int count = 0;
        for (auto& student : students) {
            if (student.Age() >= 18) {
                count++;
            }
        }
        return count;
    }

    float percentOfExcellent(){
        int excellentCount = 0;
        for (auto& student : students) {
            if (student.getMark() == 5) {
                excellentCount++;
            }
        }
        return (static_cast<float>(excellentCount) / students.size() * 100);
    }
};

int main() {
    Group group;

    group.addStudent("Земсков", 2007, 3.1);
    group.addStudent("Иванов", 2004, 2.9);
    group.addStudent("Петров", 1998, 5.0);

    group.showGroupInfo();

    std::cout << "Количество совершеннолетних в группе: "
               << group.countAdults() << std::endl;

    std::cout << "Процент отличников в группе: "
               << group.percentOfExcellent() << "%" << std::endl;

    return 0;
}

```

Запустим программу и посмотрим на вывод

```

[Running] cd "d:\Лабы\АиП\Dz_4\" && g++ main.cpp -o main && "d:\Лабы\АиП\Dz_4\"main
Информация обо всех учениках группы:
Ученик Земсков родился в 2007 году. Средняя годовая оценка: 3.1
Ученик Иванов родился в 2004 году. Средняя годовая оценка: 2.9
Ученик Петров родился в 1998 году. Средняя годовая оценка: 5
Количество совершеннолетних в группе: 2
Процент отличников в группе: 33.3333%

[Done] exited with code=0 in 0.502 seconds

```

Рисунок 3 – результат работы программы

Программа корректно работает на тестовых данных.

Составим диаграмму классов:

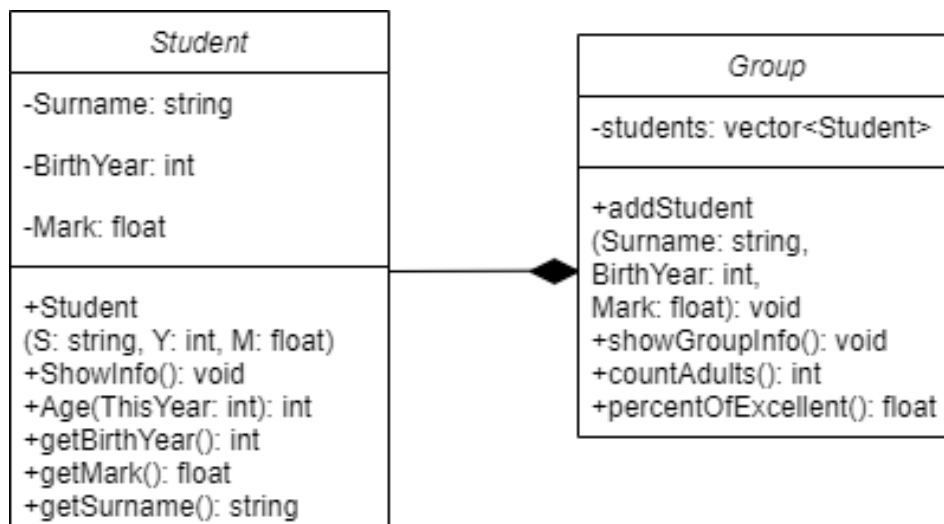


Рисунок 4 – диаграмма классов с композицией

Вывод: в ходе выполнения домашней работы научился на базовом уровне работать с массивами объектов, а также использовать композицию в написании программ.