

РТУ «МИРЭА»
КБСП

Разработка файловой системы
студентов с выполнением
задания в соответствии с
требуемым вариантом.

БАСО-06-19

Реализована загрузка данных с клавиатуры. 10

багпов

```
jdate_input:
```

```
cout << "Год поступления {ГГГГ} > ";
```

```
getline(cin, y);
```

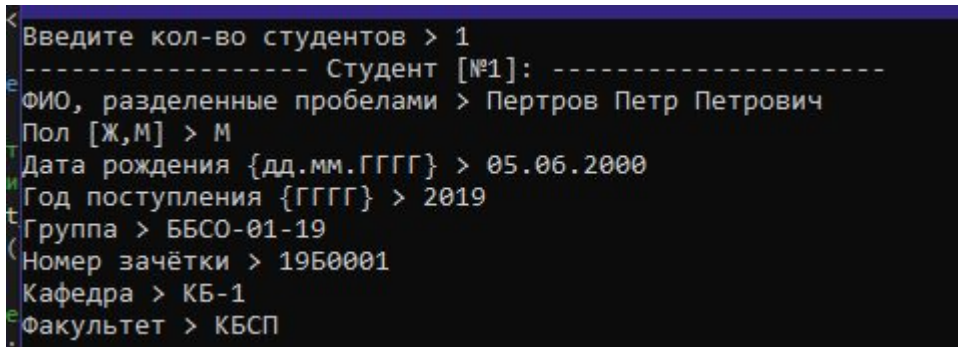
```
if (!isValidInt(y) || stoi(y) < 0 || stoi(y) > 9999 || stoi(y) <=
bday.year) {
```

```
cout << INVINP << endl;
```

```
goto jdate_input;
```

```
}
```

```
jdate.year = stoi(y);
```



```
< Введите кол-во студентов > 1
----- Студент [№1]: -----
e ФИО, разделенные пробелами > Пертров Петр Петрович
T Пол [Ж,М] > М
M Дата рождения {дд.мм.ГГГГ} > 05.06.2000
t Год поступления {ГГГГ} > 2019
( Группа > ББСО-01-19
e Номер зачётки > 19Б0001
. Кафедра > КБ-1
Факультет > КБСП
```

Программа запускается и верно
выполняет задание согласно варианту.
10 баллов.

```
void Univer::var34query(int gend) {
    for (int i = 0; i < N; i++) {
        int tc = 0;
        Student* tmpS = &((*_students)[i]);
        if (tmpS->_gender != gend) continue;
        for (int j = 0; j < tmpS->_zach->sc; j++) {
            for (int k = 0; k < tmpS->_zach->sems[j]->sc; k++) {
                if (tmpS->_zach->sems[j]->subjects[k]->mark == 3) {
                    tc++;
                    break;
                }
            }
        }
        if (tc == 0) {
            tmpS->fullIntroduce();
            tmpS->_zach->showAll();
        }
    }
}
```

Введите номер действия, которое хотите совершить:

1. Ввести/добавить студентов
2. Удалить студента
3. Показать всех студентов в университете
4. Изменить информацию о студенте
5. Показать выборку (вариант 34)
0. Выход

> 5

Введите пол, по которому необходимо производить поиск [М, Ж] > М

Фамилия: Кузнецов Имя: Валерий Отчество: Валентинович

Пол: М | День рождения: 26.11.2001

Номер зачётной книжки: 19Б3459 | Год поступления: 2018

Группа: Басо-03 | Факультет: Ик | Кафедра: 1

1 семестр:

Физика: 5

Алгебра: 5

Яп: 5

Аса: 4

Английский: 5

2 семестр:

Физика: 5

Алгебра: 5

Яп: 4

Аса: 5

Английский: 5

3 семестр:

Физика: 5

Алгебра: 5

Яп: 4

Аса: 5

Английский: 5

4 семестр:

Физика: 5

Алгебра: 5

Яп: 4

Аса: 5

Английский: 5

5 семестр:

6 семестр:

7 семестр:

8 семестр:

9 семестр:

10 семестр:

11 семестр:

12 семестр:

Фамилия: Печорин Имя: Григорий Отчество: Александрович

Пол: М | День рождения: 27.05.2002

Номер зачётной книжки: 19Б2539 | Год поступления: 2019

Группа: басо-02 | Факультет: ик | Кафедра: 1

1 семестр:

яп: 5

история: 5

аса: 5

линал: 5

матанализ: 5

2 семестр:

яп: 5

аса: 5

история: 5

линал: 5

английский: 5

3 семестр:

4 семестр:

5 семестр:

6 семестр:

7 семестр:

8 семестр:

9 семестр:

10 семестр:

11 семестр:

12 семестр:

Реализованы функции записи и чтения информации в/из файл(а). 10 баллов.

```
void Univer::loadAll()
{
    string data = readContents();
    string oneStud;
    int i = 0;
    char c = data[i];
    while (c != '\0') {
        oneStud = "";
        while (c != '\6' && c != '\0') {
            c = data[i];
            oneStud += c;
            i++;
        }
        c = data[i];
        Student* ns = new Student(oneStud);
        addStudent(ns);
    }
}
```

```
string readContents() {
    FILE *in = nullptr;

    fopen_s(&in, "db.bin", "r");
    string result = "";
    char c = '1';
    fseek(in, 0, 0);
    for (int i = 0; i < 4; i++) {
        fread(&c, 1, 1, in);
        if (c != fheader[i]) {
            cout << "Некорректный заголовок файла!" << endl;
            system("pause");
            exit(0);
        }
    }
    fseek(in, 4, 0);
    while (c != '\0') {
        fread(&c, 1, 1, in);
        result += c;
    }
    fclose(in);
    return result;
}
```

Использование динамической памяти. 5
баллов.

```
Student* is = new Student(sname, fname, mname, stoi(gender), bday, group, fac, kaf, jdate, z);  
u->addStudent(is);
```

Функция добавления или удаления записей в файле. 10 баллов

```
void StudentArray::addItem(Student* value)
{
    pCurrent = new ListItem;
    pCurrent->val = value;
    if (!length)
    {
        pStart = pCurrent;
        pCurrent->previous_item = pCurrent;
        pCurrent->next_item = pCurrent;
        pTemp = pCurrent;
    }
    else
    {
        pCurrent->previous_item = pTemp;
        pCurrent->next_item = pTemp->next_item;
        pTemp->next_item = pCurrent;
        pTemp = pCurrent;
    }
    length++;
}
```

```
void StudentArray::removeItem(string zach) {
    pCurrent = pStart;
    for (int i = 0; i <= length; i++)
    {
        if (pCurrent->val->_zach->_num == zach)
        {
            if (i == 0) {
                pStart = pCurrent->next_item;
                pCurrent->next_item->previous_item = nullptr;
            }
            else if (i == length - 1) {
                pTemp = pCurrent->previous_item;
                pCurrent->previous_item->next_item = nullptr;
            }
            else {
                pCurrent->previous_item->next_item = pCurrent->next_item;
                pCurrent->next_item->previous_item = pCurrent->previous_item;
            }
            delete pCurrent->val;
            delete pCurrent;
            length--;
            return;
        }
        else
            pCurrent = pCurrent->next_item;
    }
}
```

Функция изменения записей в файле. 10

баллов

```
getline(cin, zn);  
if (zn == "0") {  
    cls;  
    break;  
}  
if (!u->hasStudent(zn)) {  
    cout << "Не найдено студентов с таким номером зачетки, попробуйте снова [0=выход в меню] > ";  
    goto checkZn;  
}  
cls;  
u->showStudentChangeDialog(zn);  
u->saveAll();  
break;
```

Введите номер пункта, в который хотите внести изменения:

1. Фамилия (Ясенева)
2. Имя (Нина)
3. Отчество (Павловна)
4. Пол (0)
5. День рождения (30.01.2001)
6. Год поступления (2019)
7. Факультет (Ик)
8. Кафедра (1)
9. Группа (Басо-02)
10. Номер зачётки (19Б4444)
11. Предмет
0. Выйти в главное меню

> 9

Введите 0, чтобы вернуться в меню

Группа > БАСО-02

Использование конструкторов и деструкторов. 10

баллов.

```
Univer::Univer(string title) {  
    _title = title;  
    _students = new StudentArray();  
    N = 0;  
}  
Univer::~~Univer() {  
    delete _students;  
}
```


Друзья классов. 10

баллов

```
class Semestr
```

```
{
```

```
    friend class Student;
```

```
    friend class Zchetka;
```

```
    friend class Univer;
```

```
    Subj* _subjects[10];
```

```
public:
```

```
    int sc=0;
```

```
    Semestr();
```

```
    int hasSubj(string);
```

```
    Subj* getByTitle(string);
```

```
    int addSubj(Subj*);
```

```
    float average();
```

```
};
```

Наследование(простое или сложное). 10

баллов.

```
class Person {
    protected:
        string _sname;
        string _fname;
        string _mname;
        Date _bdate;
        int _gender;
public:
    void fullIntroduce();
};

class Student : public Person {
    friend class Univer;
    friend class StudentArray;
    Date _edate;
    string _group;
    string _fac;
    string _kaf;
    Zachetka* _zach;
public:
    void fullIntroduce();
    Student(string, string, string, int, Date, string, string, string, Date, Zachetka*);
    Student(string);
    ~Student();
    float fullAvg();
    string dataForExport();
};
```

Перегрузка операций. 10

баллов

```
Student & StudentArray::operator[] (const int index)
{
    if (index < 0 || index >= length)
        throw out_of_range("Индекс находится за пределами списка");
    else {
        pCurrent = pStart;
        for (int i = 0; i < index; i++)
        {
            pCurrent = pCurrent->next_item;
        }
        return *pCurrent->val;
    }
}
```