

**Московский государственный технический  
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5.

Курс «Программирование на основе классов и шаблонов»

Отчет по лабораторной работе №1  
«Обработка и печать числовой матрицы»

Выполнил:		Проверил:
студент группы ИУ5-25Б		преподаватель каф. ИУ5
Петренко Сергей		
Подпись и дата:		Подпись и дата:

г. Москва, 2018 г.

## Постановка задачи:

Разработать класс «Планета» для планет солнечной системы .

Имя планеты должно иметь тип `char*`.

Создать многофайловый проект и отладить программу, которая создает один объект класса «Планета» и выводит значения его полей на экран.

Организовать интерфейс пользователя с программой в виде меню, позволяющего выполнять следующие действия:

- чтение БД «Солнечная система» из файла; - запись БД «Солнечная система» в файл; - сортировка БД; - редактирование БД; - вывод БД на экран.

Создать текстовый файл (в блокноте) с данными о планетах солнечной системы и сохранить его в папке проекта.

Перегрузить операцию "`>>`" для класса «Планета» и *ifstream* и прочитать данные о планетах из файла в массив

«Солнечная система» из объектов класса «Планета».

Перегрузить операцию "`<<`" для классов «Планета» и *ofstream* и вывести на экран данные из массива.

Перегрузить конструктор копирования, деструктор и оператор присваивания.

Вставьте в конструкторы и деструктор печать типа «Создание (Удаление) ID *n*», где *n* - номер объекта, для которого они вызываются.

Перегрузить операции сравнения "`<`" и "`==`" для класса «Планета», используя для этого значение одного из полей.

Отсортировать массив планет солнечной системы, хранящийся в файле, с использованием перегруженных операций.

## Разработка интерфейса класса:

### Planet.h

```
#ifndef NEWLAB1_PLANET_H
#define NEWLAB1_PLANET_H
```

```
#include <iostream>
```

```
using namespace std;
```

```
class Planet;
int getSize(char*);
void edit(Planet*, int);
int find(char*, Planet*, int);
Planet* add(Planet*, Planet*, int);
Planet* del(Planet*, int, int);
int menu();
void read_db(char*, Planet*, const int);
void print_db(Planet*, int);
void write_db(char*, Planet*, int);
Planet* sort_db(Planet*, int);
```

```
//ПРОТОТИПЫ
```

```
//Объект класса планета
//Получить размер БД
//Редактирование планеты
//Поиск планеты по имени
//Создание новой планеты
//Удаление планеты
//Меню выбора
//Чтение БД
//Вывод БД
//Запись БД
//Сортировка БД
```

```
class Planet {
friend ostream& operator << (ostream &s, Planet &v);
friend istream& operator >> (istream &s, Planet &v);
friend bool operator > ( Planet &v1, Planet&v2);
```

```
//ПРОТОТИПЫ
```

```
//Перегрузка вывода
//Перегрузка ввода
//Перегрузка сравнения
```

```
//ГЕТТЕРЫ
```

```
char* getName();
int* getDiam();
bool getLife();
int getSat();
```

```
// СЕТТЕРЫ
```

```
void setName(char*);
void setDiam(int);
void setLife(bool);
void setSat(int);
```

```
// Метод проверки массива char на стертость
bool isEmpty(char*);
```

```

private:
    char* name;    //Имя
    int d;         //Диаметр
    bool life;     //Наличие жизни
    int sat;       //Количество спутников
public:
    static int count;    //Количество созданных объектов
    static int sort;     // Выбранная сортировка
    Planet(char* name, int d, bool life, int sat ){//КОНСТРУКТОР ПО ВСЕМ ПАРАМЕТРАМ
        this->name=name;
        this->d=d;
        this->life=life;
        this->sat=sat;
        count++;
        cout<<"Class Planet count:(parameters) "<<count<<endl;
    }
    Planet(){//ПУСТОЙ КОНСТРУКТОР
        count++;
        cout<<"Class Planet count:(null parameters or setter) "<<" "<<count<<endl;
    }
    Planet( Planet &planet){//КОНСТРУКТОР КОПИРОВАНИЯ
        name=planet.getName();
        d=planet.getDiam();
        life=planet.getLife();
        sat=planet.getSat();
        count++;
        cout<<"Class Planet count:(copying) "<<name<<" "<<count<<endl;
    }
    ~Planet(){//ДЕСТРУКТОР
        count--;
        name="удалено";
        if(!isEmpty(name))
            delete[] name;
        cout<<"Class Planet count:(destruct) "<<name<<" "<<count<<endl;
    }
//ГЕТТЕРЫ
    char* getName(){
        return name;
    }
    int getDiam(){
        return d;
    }
    bool getLife(){
        return life;
    }
    int getSat(){
        return sat;
    }
//СЕТТЕРЫ
    void setName(char* name){
        this->name=name;
    }
    void setDiam(int d){
        this->d=d;
    }
    void setLife(bool life){
        this->life=life;
    }
    void setSat(int sat){
        this->sat=sat;
    }
}
//ПРОВЕРКА НА СТЕПТОСТЬ
bool isEmpty(char* str){
    for(int i=0;i<20;i++){
        if(str[i]=='\0'){
            return false;
        }
    }
}

```

```

    }
}
return true;
});
#endif //NEWLAB1_PLANET_H

```

## Текст программы:

### Planet.cpp

```

#include <fstream>
#include <iostream>
#include "Planet.h"
using namespace std;

int menu() {
    cout << "\n----- МЕНЮ ----- \n";
    cout << "1 - Чтение БД      2 - Запись БД \n";
    cout << "3 - Редактировать   4 - Вывод БД \n";
    cout << "5 - Сортировка        6 - Удалить \n";
    cout << "7 - Добавить          8 - Закончить \n";
    cout << "Для выбора операции введите число от 1 до 8 \n";
    int resp;
    cin >> resp;
    cin.clear();
    cin.ignore(10, '\n');
    return resp;
}

ostream &operator<<(ostream &s, Planet &v) {
    s << v.getName() << " " << v.getDiam() << " " << v.getLife() << " " << v.getSat() << endl;
    return s;
}

istream &operator>>(istream &s, Planet &v) {
    char *name=new char[20];
    int d,sat; bool life;
    s>>name>>d>>life>>sat;
    v.setName(name);
    v.setDiam(d);
    v.setLife(life);
    v.setSat(sat);
    return s;
}

bool operator>(Planet &v1, Planet &v2) {
    switch(Planet::sort){
        case 1 ; return strcmp(v1.getName(),v2.getName())>0;
        case 2 : return v1.getDiam()>v2.getDiam();
        case 3 : return v1.getSat()>v2.getSat();
        case 4 : return strcmp(v1.getName(),v2.getName())<0;
        case 5 : return v1.getDiam()<v2.getDiam();
        case 6 : return v1.getSat()<v2.getSat();
        case 7 : return v1.getLife()>v2.getLife();
        case 8 : return v1.getLife()<v2.getLife();
        default: return strcmp(v1.getName(),v2.getName())>0;
    }
}

int getSize(char* file_name){
    ifstream in(file_name);
    int n=0;
    char* test = new char[100];
    while(!in.eof()){
        in.getline(test, 1024, '\n');
        n++;
    }
    in.close();
    return n;
}

int find(char*planet_name,Planet* bd ,int size){
    for(int i=0;i<size;i++){
        if(strcmp(planet_name,bd[i].getName())==0)

```

```

        return i;
    }
    return -1;
}

void edit(Planet* planets,int k){
    int bufint, n; bool bufbul;
    cout<<"1-name\n2-diameter\n3-life\n4-sattelites"<<endl;
    cin>>n;
    switch(n){
        case 1 :
            char buf[20];
            cin>>buf;
            strcpy(planets[k].getName(),buf);
            break;
        case 2 :
            cin>>bufint;
            planets[k].setDiam(bufint);
            break;
        case 3 :
            cin>>bufbul;
            planets[k].setLife(bufbul);
            break;
        case 4 :
            cin>>bufint;
            planets[k].setSat(bufint);
            break;
    }
}

Planet* add(Planet* planets,Planet new_planet,int size){
    Planet* new_planets = new Planet[size+1];
    for(int i = 0;i<size;i++){
        new_planets[i]=planets[i];
        planets[i].setName("удалено");
    }
    new_planets[size]=new_planet;
    delete[] planets;
    return new_planets;
}

Planet* del(Planet* planets,int size,int k){
    Planet* new_planets = new Planet[size-1];
    if(k>=0){
        for(int i=k;i<size-1;i++){
            planets[i]=planets[i+1];
        }
    }
    for(int i=0;i<size-1;i++) {
        new_planets[i] = planets[i];
        planets[i].setName("удалено");
    }
    delete[] planets;
    return new_planets;
}

void read_db(char *file_name, Planet *bd, int size) {
    ifstream in(file_name);
    for (int i = 0; i < size; i++)
        in >> bd[i];
    in.close();
}

void print_db(Planet *bd, int size) {
    for (int i = 0; i < size; i++)
        cout << bd[i];
}

void write_db(char *file_name, Planet *bd, int size) {
    ofstream out(file_name);
    for (int i = 0; i < size; i++)

```

```

        out << bd[i];
        out.close();
    }

Planet *sort_db(Planet *planets, int size) {
    Planet o;
    int h;
    for (int j = size - 1; j > 0; j--) {
        h = 0;
        for (int i = 0; i < j; i++) {
            if (planets[i] > planets[i + 1]) {
                o = planets[i];
                planets[i] = planets[i + 1];
                planets[i + 1] = o;
                h++;
            }
        }
        if (h == 0) break;
    }
    return planets;
}

```

## Main.cpp

```

#include <iostream>
#include "Planet.h"
using namespace std;
int Planet::count=0;
int Planet::sort=0;
Planet f1(Planet pl){
    pl.setName("noname");
    return pl;
}

int main() {
    cout<<"test";
}

```

```

int new_size=9;
char *file_name = "/Users/sergei/Desktop/NewLab1/bd.txt";
new_size = getSize(file_name) - 1;
Planet *planets = new Planet[new_size];
Planet planet;
while (true) {
    switch (menu()) {
        case 1:
            read_db(file_name, planets, new_size);
            break;
        case 2:
            write_db(file_name, planets, new_size);
            break;
        case 3:
            char buf[20];
            int k;
            cout << "Какую планету редактируем?" << endl;
            cin >> buf;
            cout << endl;
            k = find(buf, planets, new_size);
            if (k >= 0) {
                edit(planets, k);
            } else
                cout << "Такой планеты нет в БД" << endl;
            break;
        case 4:
            print_db(planets, new_size);
            break;
        case 5: cout<<"\n 1-nameUP \n 2-DiamUP \n 3-SatUP \n 4-nameDown \n 5-DiamDown \n
6-SatDown \n 7-LifeUp \n 8-LifeDown \n";
            cin>>Planet::sort;
            planets = sort_db(planets, new_size);
            break;
        case 6:
            cout << "Какую планету удаляем?" << endl;
            cin >> buf;
            k = find(buf, planets, new_size);
            if (k >= 0) {
                planets = del(planets, new_size, k);
                new_size--;
            }
            break;
        case 7:
            cout << "Введите новые данные:" << endl;
            cin >> planet;
            planets = add(planets, planet, new_size);
            new_size++;
            break;
        case 8:
            planets[1]=f1(planets[0]);
            delete[] planets;
            return 0;
        default:
            cout << " Неправильный ввод" << endl;
            break;
    }
}
}
}

```

## Анализ результатов:

```
testClass Planet count:(null parameters or setter) 1
Class Planet count:(null parameters or setter) 2
Class Planet count:(null parameters or setter) 3
Class Planet count:(null parameters or setter) 4
Class Planet count:(null parameters or setter) 5
Class Planet count:(null parameters or setter) 6
Class Planet count:(null parameters or setter) 7
Class Planet count:(null parameters or setter) 8
Class Planet count:(null parameters or setter) 9
Class Planet count:(null parameters or setter) 10
Class Planet count:(null parameters or setter) 11

----- МЕНЮ -----
1 - Чтение БД      2 - Запись БД
3 - Редактировать  4 - Вывод БД
5 - Сортировка    6 - Удалить
7 - Добавить      8 - Закончить
Для выбора операции введите число от 1 до 8
1

----- МЕНЮ -----
1 - Чтение БД      2 - Запись БД
3 - Редактировать  4 - Вывод БД
5 - Сортировка    6 - Удалить
7 - Добавить      8 - Закончить
Для выбора операции введите число от 1 до 8
4
Earth 12774 1 1
Jupiter 142796 0 16
Mars 6786 1 2
Mercury 4878 0 0
Neptune 49600 0 2
Ppp 123 1 1
Saturn 120000 0 17
Uranus 51108 0 5
Venus 12104 0 0
aaa 1 1 1

----- МЕНЮ -----
1 - Чтение БД      2 - Запись БД
3 - Редактировать  4 - Вывод БД
5 - Сортировка    6 - Удалить
7 - Добавить      8 - Закончить
Для выбора операции введите число от 1 до 8
```

```
Для выбора операции введите число от 1 до 8
7
Введите новые данные:
Bbb 1 1 1
Class Planet count:(copying) Bbb 11
Class Planet count:(null parameters or setter) 12
Class Planet count:(null parameters or setter) 13
Class Planet count:(null parameters or setter) 14
Class Planet count:(null parameters or setter) 15
Class Planet count:(null parameters or setter) 16
Class Planet count:(null parameters or setter) 17
Class Planet count:(null parameters or setter) 18
Class Planet count:(null parameters or setter) 19
Class Planet count:(null parameters or setter) 20
Class Planet count:(null parameters or setter) 21
Class Planet count:(destruct) удалено 20
Class Planet count:(destruct) удалено 19
Class Planet count:(destruct) удалено 18
Class Planet count:(destruct) удалено 17
Class Planet count:(destruct) удалено 16
Class Planet count:(destruct) удалено 15
Class Planet count:(destruct) удалено 14
Class Planet count:(destruct) удалено 13
Class Planet count:(destruct) удалено 12
Class Planet count:(destruct) удалено 11

----- МЕНЮ -----
1 - Чтение БД      2 - Запись БД
3 - Редактировать  4 - Вывод БД
5 - Сортировка    6 - Удалить
7 - Добавить      8 - Закончить
Для выбора операции введите число от 1 до 8
4
Earth 12774 1 1
Jupiter 142796 0 16
Mars 6786 1 2
Mercury 4878 0 0
Neptune 49600 0 2
Ppp 123 1 1
Saturn 120000 0 17
Uranus 51108 0 5
Venus 12104 0 0
```

```
Для выбора операции введите число от 1 до 8
6
Какую планету удаляем?
aaa
Class Planet count:(null parameters or setter) 12
Class Planet count:(null parameters or setter) 13
Class Planet count:(null parameters or setter) 14
Class Planet count:(null parameters or setter) 15
Class Planet count:(null parameters or setter) 16
Class Planet count:(null parameters or setter) 17
Class Planet count:(null parameters or setter) 18
Class Planet count:(null parameters or setter) 19
Class Planet count:(null parameters or setter) 20
Class Planet count:(destruct) удалено 19
Class Planet count:(destruct) удалено 18
Class Planet count:(destruct) удалено 17
Class Planet count:(destruct) удалено 16
Class Planet count:(destruct) удалено 15
Class Planet count:(destruct) удалено 14
Class Planet count:(destruct) удалено 13
Class Planet count:(destruct) удалено 12
Class Planet count:(destruct) удалено 11
Class Planet count:(destruct) удалено 10

----- МЕНЮ -----
1 - Чтение БД      2 - Запись БД
3 - Редактировать  4 - Вывод БД
5 - Сортировка    6 - Удалить
7 - Добавить      8 - Закончить
Для выбора операции введите число от 1 до 8
```

```
----- МЕНЮ -----
1 - Чтение БД      2 - Запись БД
3 - Редактировать  4 - Вывод БД
5 - Сортировка    6 - Удалить
7 - Добавить      8 - Закончить
Для выбора операции введите число от 1 до 8
3
Какую планету редактируем?
Mars
1-name
2-diameter
3-life
4-sattelites
1
NewMars

----- МЕНЮ -----
1 - Чтение БД      2 - Запись БД
3 - Редактировать  4 - Вывод БД
5 - Сортировка    6 - Удалить
7 - Добавить      8 - Закончить
Для выбора операции введите число от 1 до 8
4
Earth 12774 1 1
Jupiter 142796 0 16
NewMars 6786 1 2
Mercury 4878 0 0
Neptune 49600 0 2
Ppp 123 1 1
Saturn 120000 0 17
Uranus 51108 0 5
Venus 12104 0 0
Bbb 1 1 1

----- МЕНЮ -----
```



```

----- МЕНЮ -----
1 - Чтение БД          2 - Запись БД
3 - Редактировать      4 - Вывод БД
5 - Сортировка         6 - Удалить
7 - Добавить          8 - Закончить
Для выбора операции введите число от 1 до 8
8
Class Planet count:(copying) Bbb 12
Class Planet count:(copying) noname 13
Class Planet count:(destruct) удалено 12
Class Planet count:(destruct) удалено 11
Class Planet count:(destruct) удалено 10
Class Planet count:(destruct) удалено 9
Class Planet count:(destruct) удалено 8
Class Planet count:(destruct) удалено 7
Class Planet count:(destruct) удалено 6
Class Planet count:(destruct) удалено 5
Class Planet count:(destruct) удалено 4
Class Planet count:(destruct) удалено 3
Class Planet count:(destruct) удалено 2
Class Planet count:(destruct) удалено 1
Class Planet count:(destruct) удалено 0

Process finished with exit code 0

```

```

Для выбора операции введите число от 1 до 8
5
1-nameUP
2-DiamUP
3-SatUP
4-nameDown
5-DiamDown
6-SatDown
7-LifeUp
8-LifeDown
1
Class Planet count:(null parameters or setter) 11
Class Planet count:(destruct) удалено 10

----- МЕНЮ -----
1 - Чтение БД          2 - Запись БД
3 - Редактировать      4 - Вывод БД
5 - Сортировка         6 - Удалить
7 - Добавить          8 - Закончить
Для выбора операции введите число от 1 до 8
4
Earth 12774 1 1
Jupiter 142796 0 16
Mars 6786 1 2
Mercury 4878 0 0
Neptune 49600 0 2
Saturn 120000 0 17
Uranus 51108 0 5
Venus 12104 0 0

```