**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Программирование на основе классов и шаблонов»

Отчет по лабораторной работе №1

«Планеты»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-21 |  | преподаватель каф. ИУ5 |
| Михалёв Я.М. |  | Козлов А.Д. |
| Подпись и дата: |  | Подпись и дата: |
|  |  |  |

Москва, 2022 г.

Постановка задачи

Массив классов планет

1) Ввести данные планеты (переназвать планеты с одним именем)

2) Отсортировать планеты по названию, диаметру, количеству спутников

3) Вывести информацию о планете по имени

4) Поменять данные планеты (минимум написать только переименование, новое имя не должно быть уже имеющимся)

5) Удалить планету

6) Обязательно функция поиска планеты по имени

7) Вывод всех планет

Разработка алгоритма

**Класс Planets**

Свойства:

* string name – название планеты
* long int diam – диаметр планеты
* int sat – количество спутников у планеты
* string life – есть ли жизнь на планете?

Методы:

* GetName()
* GetDiam()
* GetSat()
* GetLife()
* SetName(string newName)
* SetDiam(long int newDiam)
* SetSat(int newSat)
* SetLife(bool newLife)

**Входные данные**

* Массив планет (название, диаметр, количество спутников, жизнь)

**Меню программы**

* Добавление планеты
* Удаление планеты
* Вывод всех планет
* Вывод одной планеты
* Изменение названия планеты
* Сортировка планет по названию
* Сортировка планет по диаметру
* Сортировка планет по количеству спутников
* Выход из программы

**Список функций**

**AddPlanet**

Входные данные

* string name – название планеты
* long int diam – диаметр планеты
* int sat – количество спутников
* bool life – есть ли жизнь на планете

Ограничения на ввод

* name – ненулевая строка
* diam – положительное целое число
* sat – неотрицательное целое число
* life – целое число 0 или 1

Назначение

Добавляет в массив планет planets новую планету

**RemovePlanet**

Входные данные

* string name – название планеты (ненулевая строка)

Назначение

Удаляет планету. Если планеты с названием name не существует, то выводит сообщение об ошибке

**PrintPlanet**

Входные данные

* string name – название планеты (ненулевая строка)

Выходные данные

* string name – название планеты
* long int diam – диаметр планеты
* int sat – количество спутников
* string life – есть ли жизнь на планете

Назначение

* Выводит данные о планете с названием name

**PrintPlanets**

Назначение

* Выводит информацию о всех планетах

**Rename**

Входные данные

* string name – старое название планеты
* string newName – новое название планеты

Ограничения на ввод

* name – ненулевая строка
* newName – ненулевая строка
* name != newName
* Должна существовать планета с названием name
* Не должно существовать планеты с названием newName

Назначение

* Переименовывает планету с названием name в newName

**SortByName**

Входные данные

* bool flag – сортировка по возрастанию / убыванию

Назначение

Сортирует пузырьком список планет planets по возрастанию / убыванию названий в зависимости от переменной flag

**SortByDiam**

Входные данные

* bool flag – сортировка по возрастанию / убыванию

Назначение

Сортирует пузырьком список планет planets по возрастанию / убыванию диаметра в зависимости от переменной flag

**SortBySat**

Входные данные

* bool flag – сортировка по возрастанию / убыванию

Назначение

Сортирует пузырьком список планет planets по возрастанию / убыванию количества спутников в зависимости от переменной flag

**Menu**

Входные данные

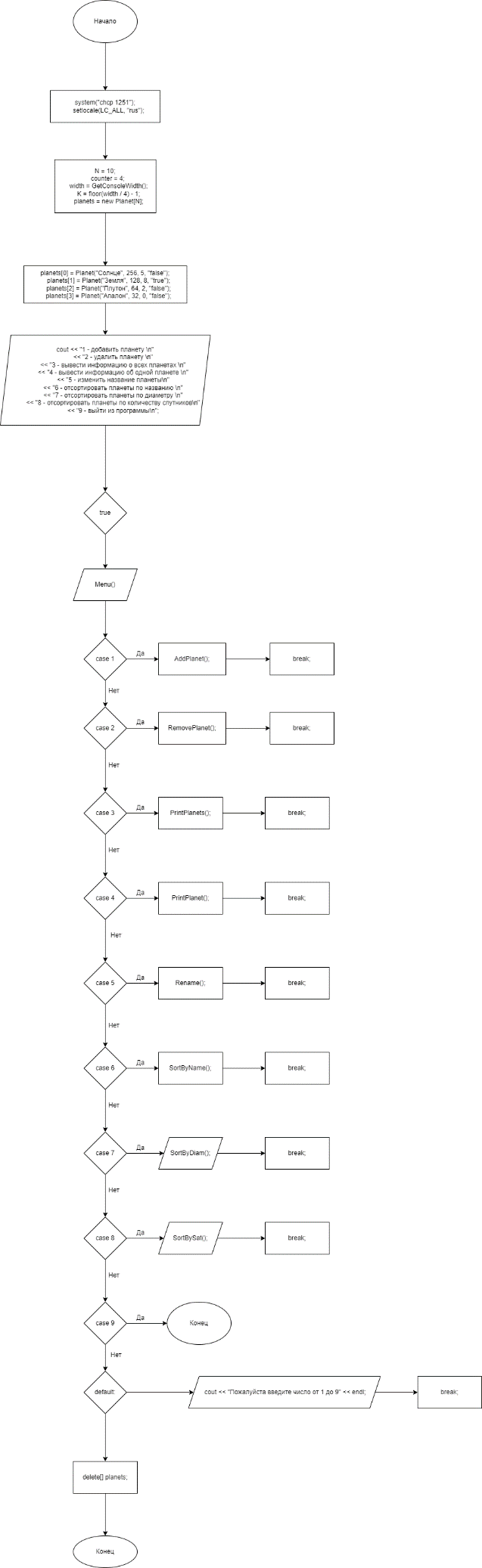
* int num – число от 1 до 9

Назначение

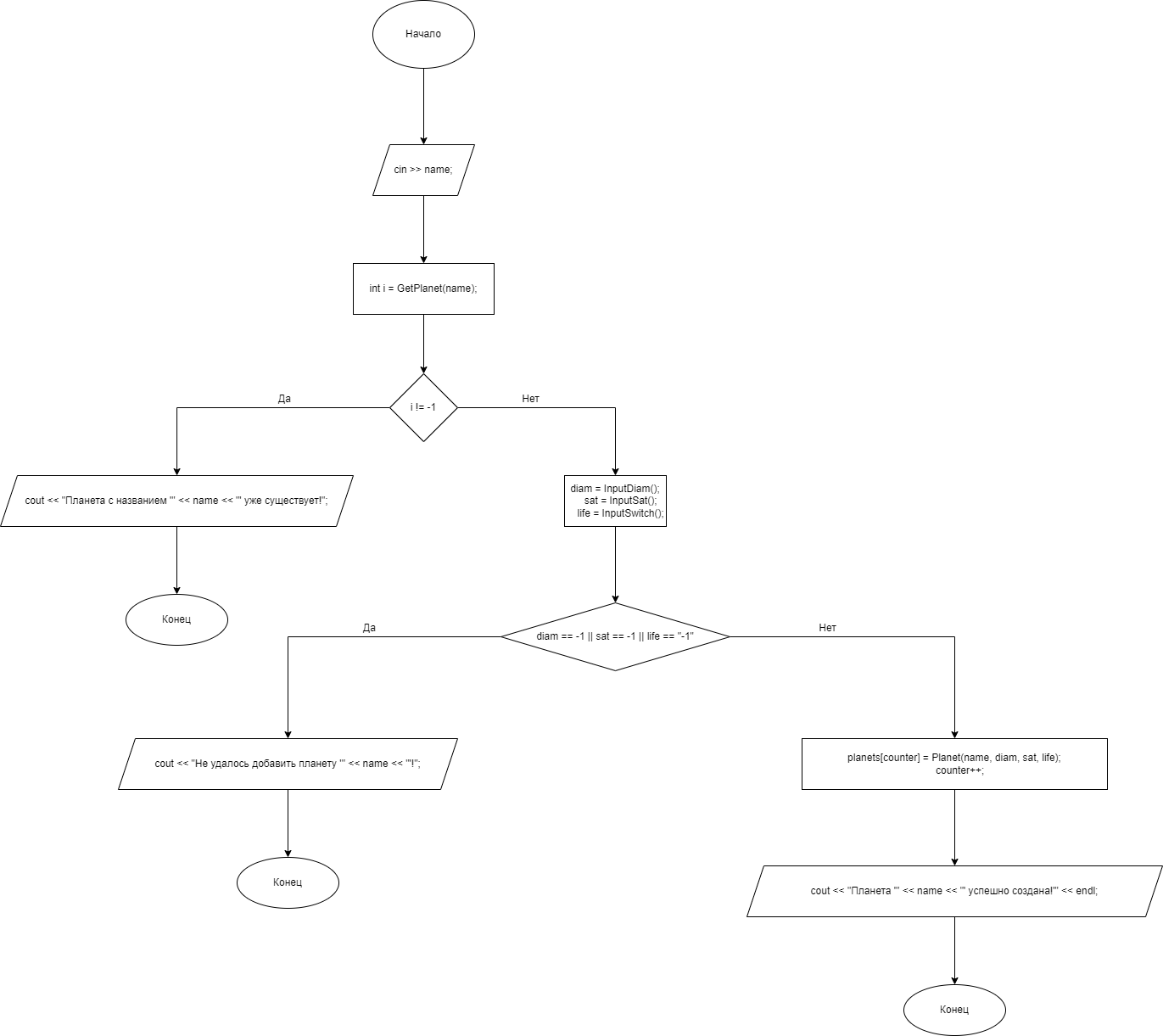
Отвечает за взаимодействие между пользователем и программой

**Схема алгоритма**

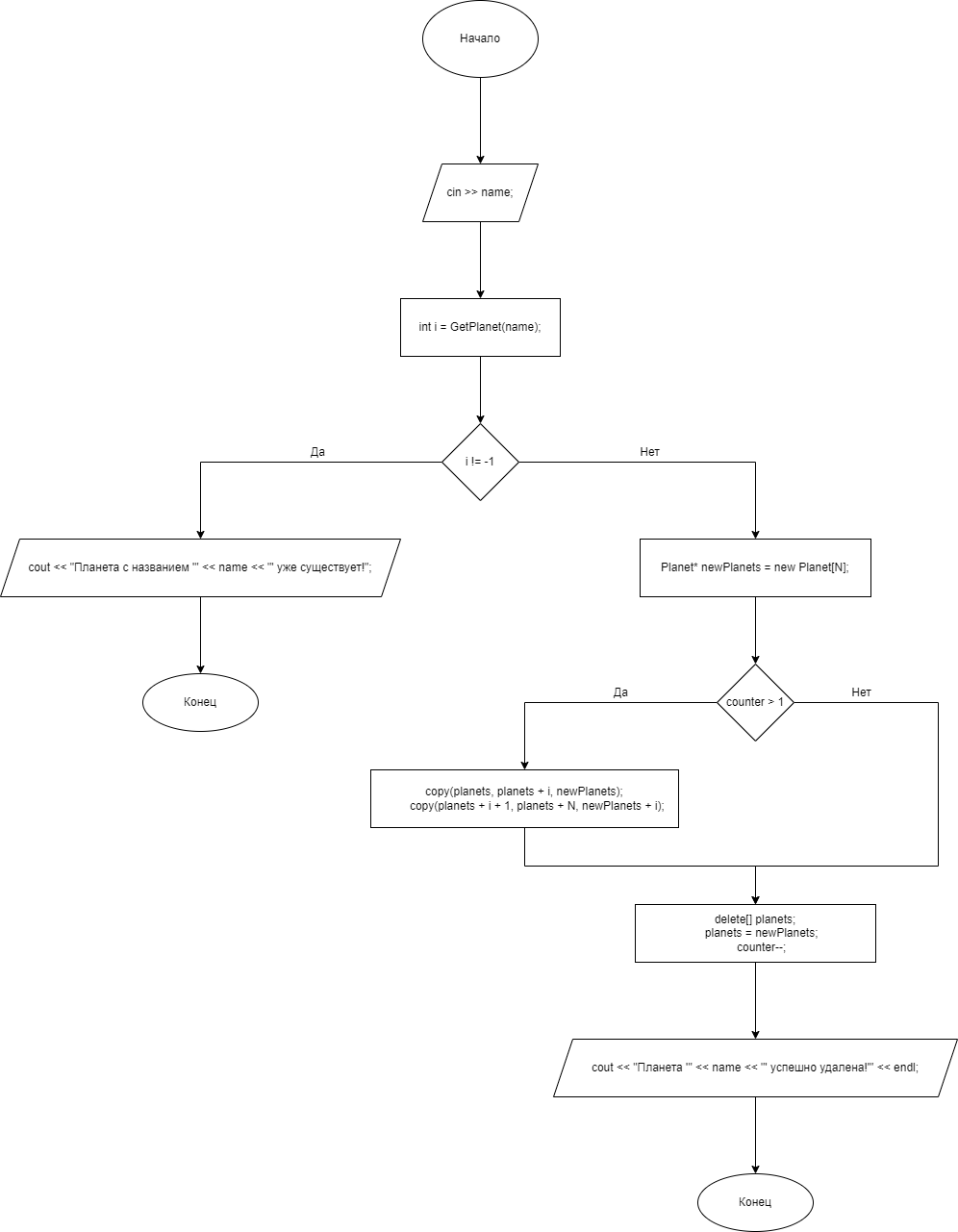
main

****

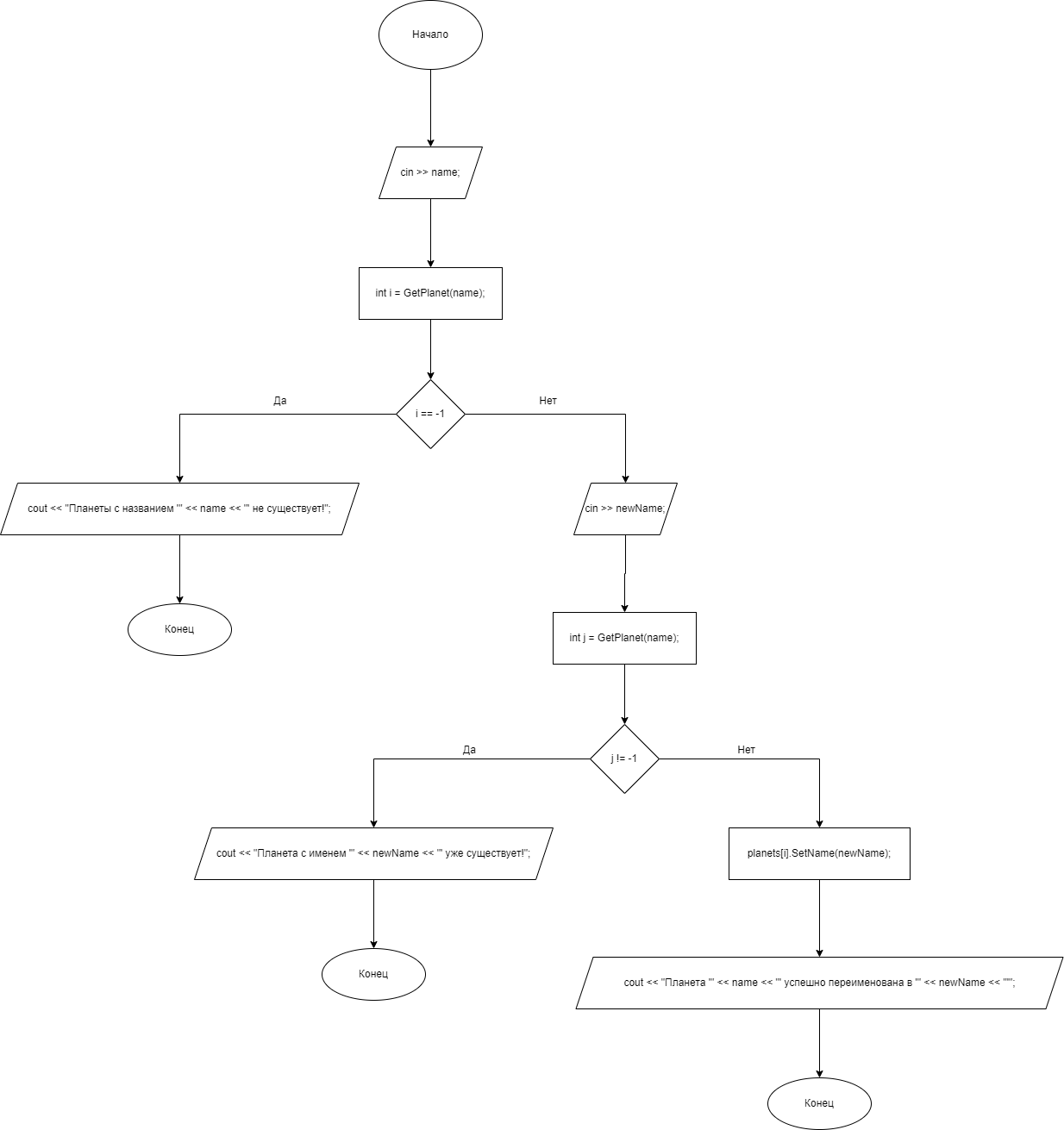
AddPlanet



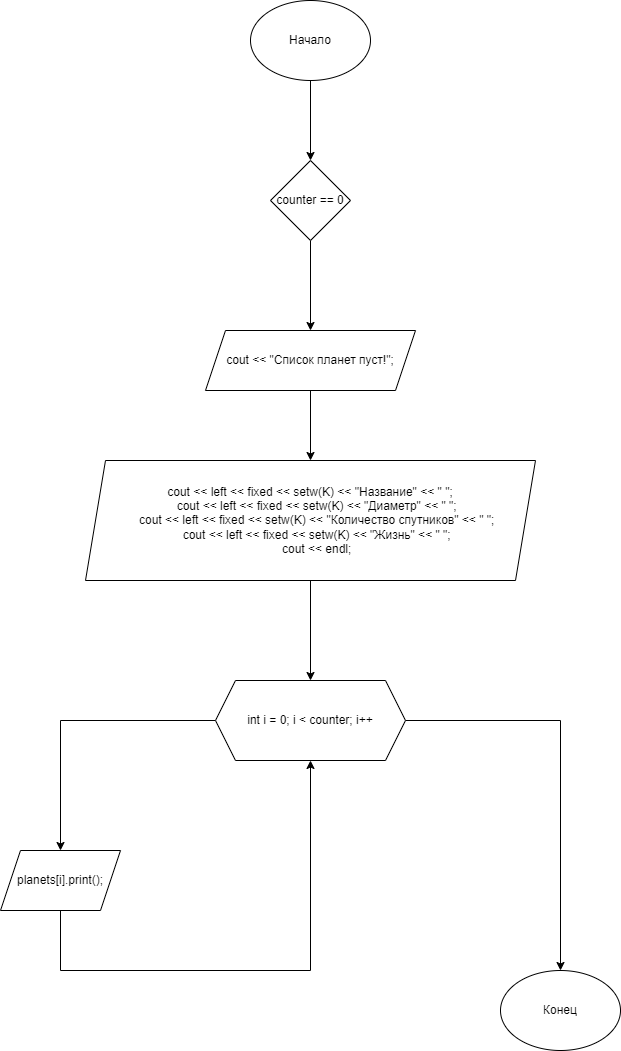
RemovePlanet



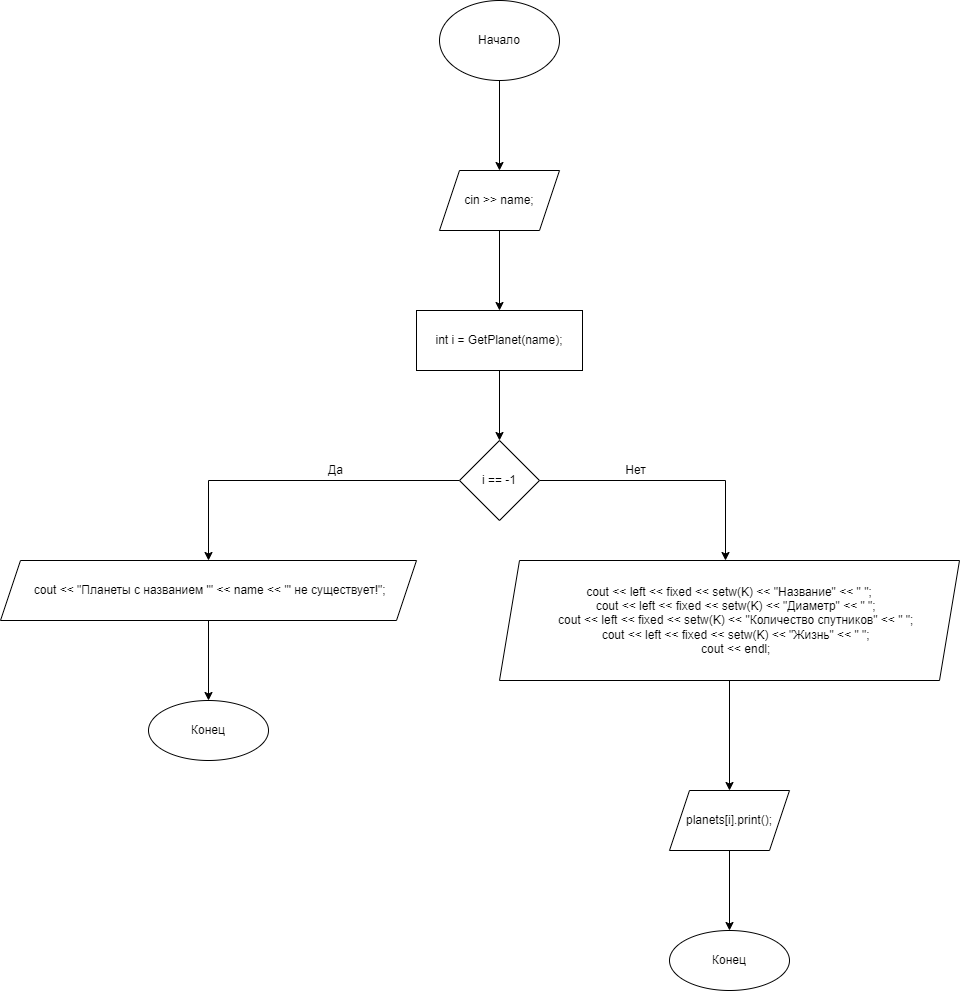
Rename



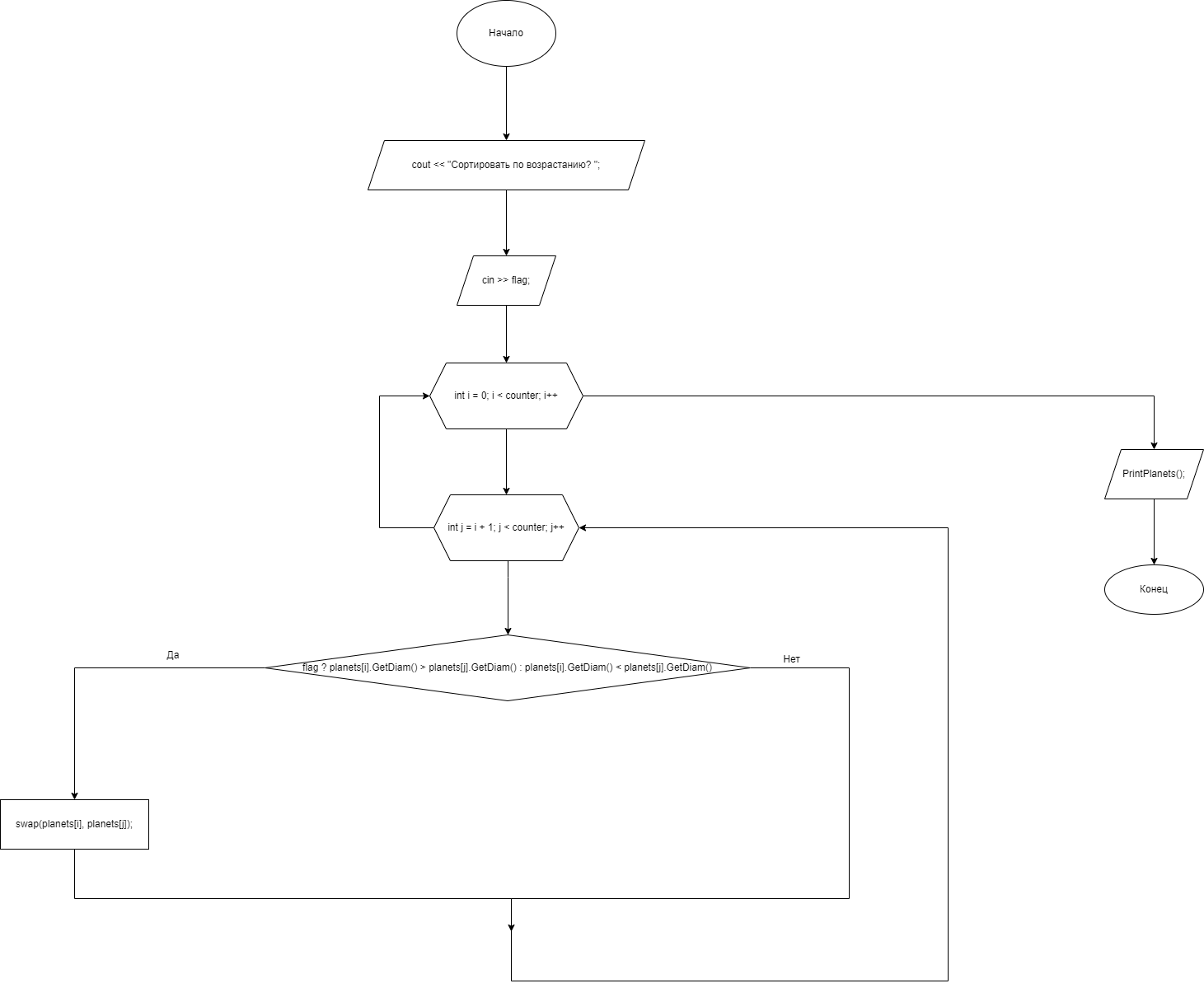
PrintPlanets



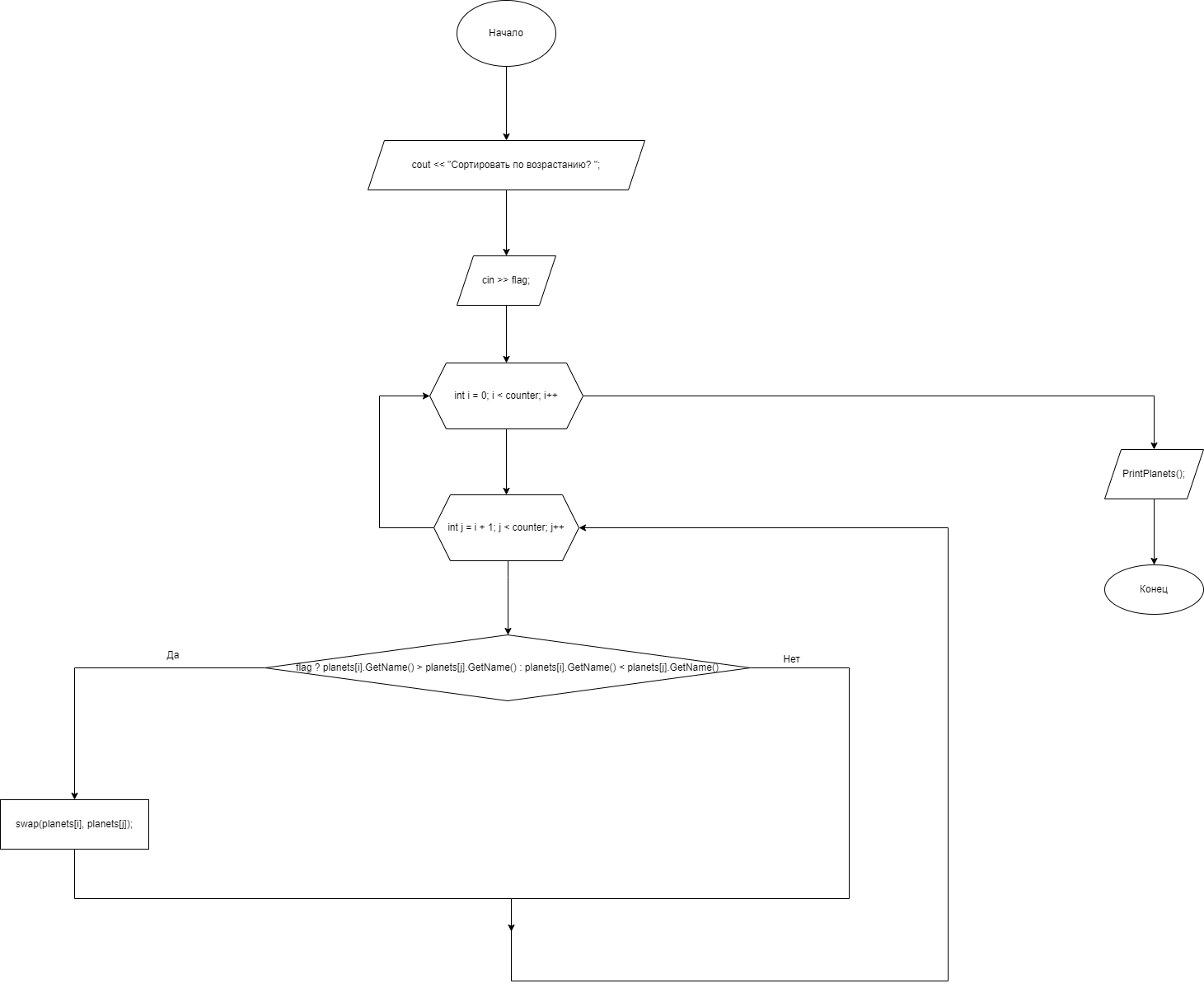
PrintPlanet



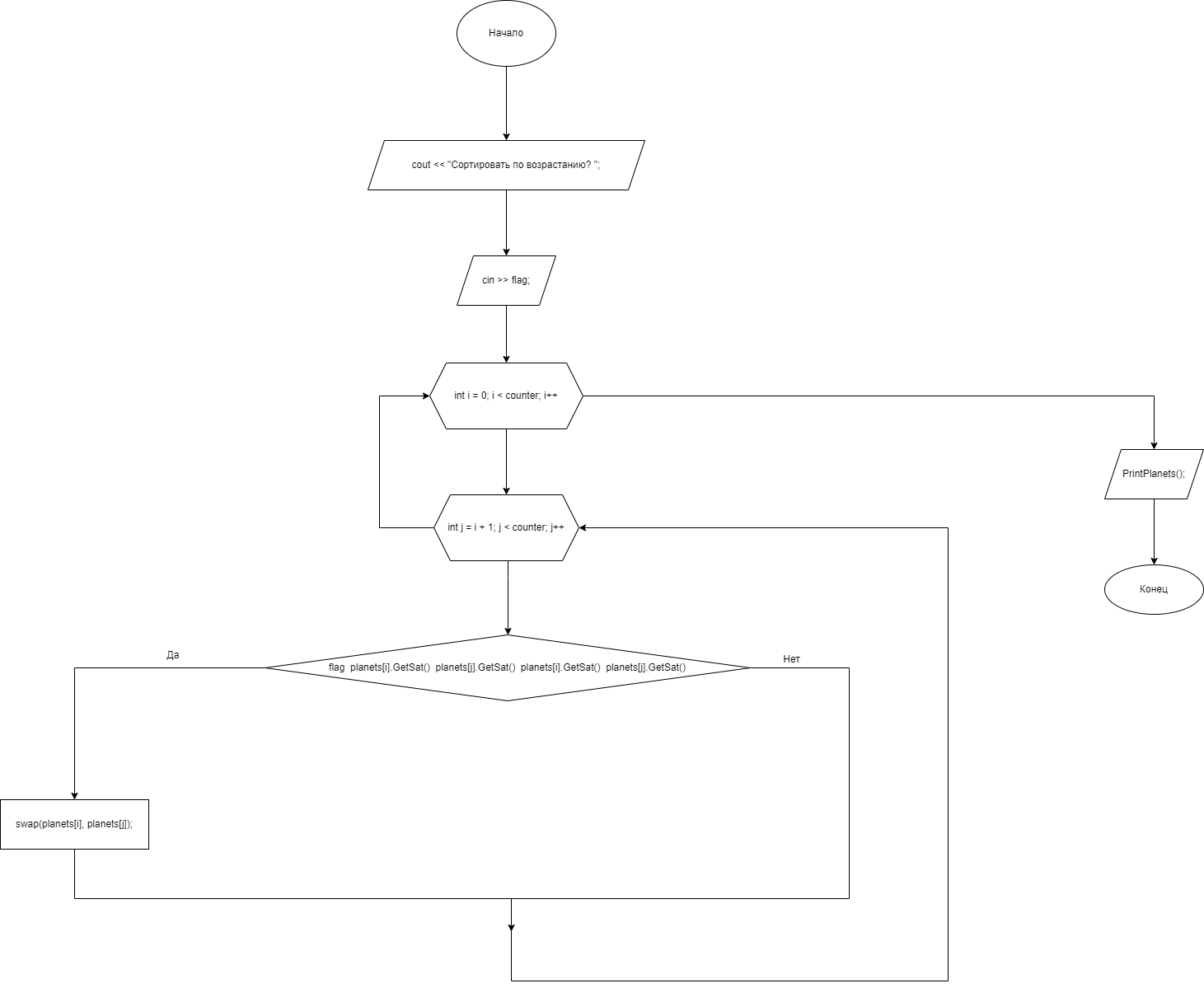
SortByDiam



SortByName



SortBySat



Текст программы

Main.cpp

#include <iostream>

#include <string>

#include <iomanip>

#include "header.h"

**using** **namespace** std**;**

int N**;**

int K**;**

int width**;**

int counter**;**

class Planet **{**

private**:**

string name**;**

long int diam**;**

int sat**;**

string life**;**

public**:**

Planet**()**

**{**

name **=** "planet\_name"**;**

diam **=** 123**;**

sat **=** 12**;**

life **=** **true;**

**}**

Planet**(**string name**,** long int diam**,** int sat**,** string life**)**

**{**

**this->**name **=** name**;**

**this->**diam **=** diam**;**

**this->**sat **=** sat**;**

**this->**life **=** life**;**

**}**

void print**()**

**{**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** **this->**name **<<** " "**;**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** **this->**diam **<<** " "**;**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** **this->**sat **<<** " "**;**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** **this->**life **<<** " "**;**

cout **<<** endl**;**

**}**

inline string GetName**()** **{** **return** **this->**name**;** **}**

inline long int GetDiam**()** **{** **return** **this->**diam**;** **}**

inline int GetSat**()** **{** **return** **this->**sat**;** **}**

inline string GetLife**()** **{** **return** **this->**life**;** **}**

inline void SetName**(**string newName**)** **{** **this->**name **=** newName**;** **};**

inline void SetDiam**(**long int newDiam**)** **{** **this->**diam **=** newDiam**;** **};**

inline void SetSat**(**int newSat**)** **{** **this->**sat **=** newSat**;** **};**

inline void SetLife**(**bool newLife**)** **{** **this->**life **=** newLife**;** **};**

**~**Planet**()**

**{**

**}**

**};**

Planet**\*** planets **=** **nullptr;**

int GetPlanet**(**string name**)**

**{**

**for** **(**int i **=** 0**;** i **<** counter**;** i**++)**

**{**

**if** **(**planets**[**i**].**GetName**()** **==** name**)**

**{**

**return** i**;**

**}**

**}**

**return** **-**1**;**

**}**

void AddPlanet**()**

**{**

string name**;**

long int diam**;**

int sat**;**

string life**;**

cout **<<** "Введите название планеты\n> "**;**

cin **>>** name**;**

int i **=** GetPlanet**(**name**);**

**if** **(**i **!=** **-**1**)**

**{**

cout **<<** "Планета с названием '" **<<** name **<<** "' уже существует!"**;**

**return;**

**}**

diam **=** InputDiam**();**

sat **=** InputSat**();**

life **=** InputSwitch**();**

**if** **(**diam **==** **-**1 **||** sat **==** **-**1 **||** life **==** "-1"**)**

**{**

cout **<<** "Не удалось добавить планету '" **<<** name **<<** "'!"**;**

**return;**

**}**

planets**[**counter**]** **=** Planet**(**name**,** diam**,** sat**,** life**);**

counter**++;**

cout **<<** "Планета '" **<<** name **<<** "' успешно создана!'" **<<** endl**;**

**}**

void RemovePlanet**()** **{**

string name**;**

cout **<<** "Введите название планеты\n> "**;**

cin **>>** name**;**

int i **=** GetPlanet**(**name**);**

**if** **(**i **==** **-**1**)**

**{**

cout **<<** "Планеты с названием '" **<<** name **<<** "' не существует!"**;**

**return;**

**};**

Planet**\*** newPlanets **=** **new** Planet**[**N**];**

**if** **(**counter **>** 1**)**

**{**

copy**(**planets**,** planets **+** i**,** newPlanets**);**

copy**(**planets **+** i **+** 1**,** planets **+** N**,** newPlanets **+** i**);**

**}**

**delete[]** planets**;**

planets **=** newPlanets**;**

counter**--;**

cout **<<** "Планета '" **<<** name **<<** "' успешно удалена!'" **<<** endl**;**

**return;**

**}**

void Rename**()**

**{**

string name**;**

string newName**;**

cout **<<** "Введите название планеты, которую хотите переименовать\n> "**;**

cin **>>** name**;**

int i **=** GetPlanet**(**name**);**

**if** **(**i **==** **-**1**)**

**{**

cout **<<** "Планеты с названием '" **<<** name **<<** "' не существует!"**;**

**return;**

**}**

cout **<<** "Введите новое название для планеты '" **<<** name **<<** "'\n> "**;**

cin **>>** newName**;**

int j **=** GetPlanet**(**newName**);**

**if** **(**j **!=** **-**1**)**

**{**

cout **<<** "Планета с именем '" **<<** newName **<<** "' уже существует!"**;**

**return;**

**}**

planets**[**i**].**SetName**(**newName**);**

cout **<<** "Планета '" **<<** name **<<** "' успешно переименована в '" **<<** newName **<<** "'"**;**

**}**

void PrintPlanets**()**

**{**

**if** **(**counter **==** 0**)**

**{**

cout **<<** "Список планет пуст!"**;**

**return;**

**}**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** "Название" **<<** " "**;**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** "Диаметр" **<<** " "**;**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** "Количество спутников" **<<** " "**;**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** "Жизнь" **<<** " "**;**

cout **<<** endl**;**

**for** **(**int i **=** 0**;** i **<** counter**;** i**++)**

**{**

planets**[**i**].**print**();**

**}**

**}**

void PrintPlanet**()**

**{**

string name**;**

cout **<<** "Введите название планеты\n> "**;**

cin **>>** name**;**

int i **=** GetPlanet**(**name**);**

**if** **(**i **==** **-**1**)**

**{**

cout **<<** "Планеты с названием '" **<<** name **<<** "' не существует!"**;**

**return;**

**}**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** "Название" **<<** " "**;**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** "Диаметр" **<<** " "**;**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** "Количество спутников" **<<** " "**;**

cout **<<** left **<<** fixed **<<** setw**(**K**)** **<<** "Жизнь" **<<** " "**;**

cout **<<** endl**;**

planets**[**i**].**print**();**

**}**

void SortByName**()**

**{**

bool flag**;**

cout **<<** "Сортировать по возрастанию? 1/0\n> "**;**

cin **>>** flag**;**

**for** **(**int i **=** 0**;** i **<** counter**;** i**++)**

**{**

**for** **(**int j **=** i **+** 1**;** j **<** counter**;** j**++)**

**{**

**if** **(**flag **?** planets**[**i**].**GetName**()** **>** planets**[**j**].**GetName**()** **:** planets**[**i**].**GetName**()** **<** planets**[**j**].**GetName**())**

swap**(**planets**[**i**],** planets**[**j**]);**

**}**

**}**

PrintPlanets**();**

**}**

void SortByDiam**()**

**{**

bool flag**;**

cout **<<** "Сортировать по возрастанию? 1/0\n> "**;**

cin **>>** flag**;**

**for** **(**int i **=** 0**;** i **<** counter**;** i**++)**

**{**

**for** **(**int j **=** i **+** 1**;** j **<** counter**;** j**++)**

**{**

**if** **(**flag **?** planets**[**i**].**GetDiam**()** **>** planets**[**j**].**GetDiam**()** **:** planets**[**i**].**GetDiam**()** **<** planets**[**j**].**GetDiam**())**

swap**(**planets**[**i**],** planets**[**j**]);**

**}**

**}**

PrintPlanets**();**

**}**

void SortBySat**()**

**{**

bool flag**;**

cout **<<** "Сортировать по возрастанию? 1/0\n> "**;**

cin **>>** flag**;**

**for** **(**int i **=** 0**;** i **<** counter**;** i**++)**

**{**

**for** **(**int j **=** i **+** 1**;** j **<** counter**;** j**++)**

**{**

**if** **(**flag **?** planets**[**i**].**GetSat**()** **>** planets**[**j**].**GetSat**()** **:** planets**[**i**].**GetSat**()** **<** planets**[**j**].**GetSat**())**

swap**(**planets**[**i**],** planets**[**j**]);**

**}**

**}**

PrintPlanets**();**

**}**

int menu**()**

**{**

int choose**;**

cout **<<** "\n\n\nВведите число от 1 до 9: "**;**

**if** **(**cin **>>** choose**)**

**{**

**}**

**else**

**{**

cin**.**clear**();** // put the stream back into a good state

cin**.**ignore**(**1000**,** '\n'**);** // and remove the bad input remaining in the input buffer

**}**

**return** choose**;**

**}**

int main**()**

**{**

system**(**"chcp 1251"**);**

setlocale**(**LC\_ALL**,** "rus"**);**

// cout << "Введите количество планет\n> ";

// cin >> N;

N **=** 10**;**

counter **=** 4**;**

width **=** GetConsoleWidth**();**

K **=** floor**(**width **/** 4**)** **-** 1**;**

planets **=** **new** Planet**[**N**];**

planets**[**0**]** **=** Planet**(**"Солнце"**,** 256**,** 5**,** "false"**);**

planets**[**1**]** **=** Planet**(**"Земля"**,** 128**,** 8**,** "true"**);**

planets**[**2**]** **=** Planet**(**"Плутон"**,** 64**,** 2**,** "false"**);**

planets**[**3**]** **=** Planet**(**"Апалон"**,** 32**,** 0**,** "false"**);**

cout **<<** "1 - добавить планету \n"

**<<** "2 - удалить планету \n"

**<<** "3 - вывести информацию о всех планетах \n"

**<<** "4 - вывести информацию об одной планете \n"

**<<** "5 - изменить название планеты\n"

**<<** "6 - отсортировать планеты по названию \n"

**<<** "7 - отсортировать планеты по диаметру \n"

**<<** "8 - отсортировать планеты по количеству спутников\n"

**<<** "9 - выйти из программы\n"**;**

**while** **(true)** **{**

**switch** **(**menu**())** **{**

**case** 1**:** AddPlanet**();** **break;**

**case** 2**:** RemovePlanet**();** **break;**

**case** 3**:** PrintPlanets**();** **break;**

**case** 4**:** PrintPlanet**();** **break;**

**case** 5**:** Rename**();** **break;**

**case** 6**:** SortByName**();** **break;**

**case** 7**:** SortByDiam**();** **break;**

**case** 8**:** SortBySat**();** **break;**

**case** 9**:** **return** 0**;**

**default:**

cout **<<** "Пожалуйста введите число от 1 до 9" **<<** endl**;**

**break;**

**}**

**}**

**delete[]** planets**;**

**return** 0**;**

**}**

Utils.cpp

#include <iostream>

#include "header.h"

#include <Windows.h>

**using** **namespace** std**;**

// Вовращает ширину консоли

int GetConsoleWidth**()**

**{**

COORD**\*** Chars**;**

HANDLE hCon **=** GetStdHandle**(**STD\_OUTPUT\_HANDLE**);**

CONSOLE\_SCREEN\_BUFFER\_INFO csbInfo**;**

GetConsoleScreenBufferInfo**(**hCon**,** **&**csbInfo**);**

int width **=** csbInfo**.**srWindow**.**Right **-** csbInfo**.**srWindow**.**Left **+** 1**;**

**return** width**;**

**}**

long int InputDiam**()**

**{**

**if** **(**cin**.**fail**())**

**{**

**return** **-**1**;**

**}**

long int diam**;**

cout **<<** "Введите диаметр планеты\n> "**;**

cin **>>** diam**;**

**if** **(**cin**.**fail**())**

**{**

**return** **-**1**;**

**}**

**while** **(**diam **<=** 0**)**

**{**

cout **<<** "Пожалуйста введите положительное число!\n> "**;**

cin **>>** diam**;**

**}**

**return** diam**;**

**}**

int InputSat**()**

**{**

**if** **(**cin**.**fail**())**

**{**

**return** **-**1**;**

**}**

int diam**;**

cout **<<** "Введите количество спутников\n> "**;**

cin **>>** diam**;**

**if** **(**cin**.**fail**())**

**{**

**return** **-**1**;**

**}**

**while** **(**diam **<** 0**)**

**{**

cout **<<** "Пожалуйста введите неотрицательное число!\n> "**;**

cin **>>** diam**;**

**}**

**return** diam**;**

**}**

string InputSwitch**()**

**{**

**if** **(**cin**.**fail**())**

**{**

**return** "-1"**;**

**}**

int life**;**

cout **<<** "На планете есть жизнь? 1/0\n> "**;**

cin **>>** life**;**

**if** **(**cin**.**fail**())**

**{**

**return** "-1"**;**

**}**

**while** **(**life **<** 0 **||** life **>** 1**)**

**{**

cout **<<** "Пожалуйста введите 0 или 1!\n> "**;**

cin **>>** life**;**

**}**

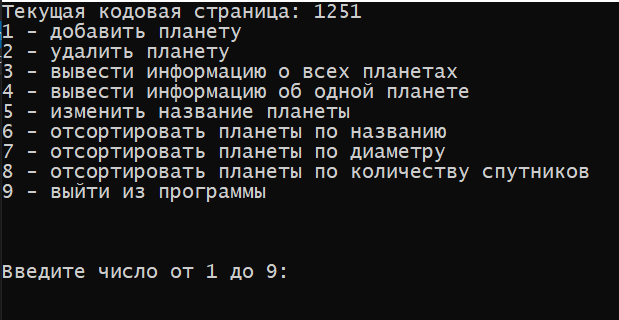
**if** **(**life**)** **return** "true"**;**

**return** "false"**;**

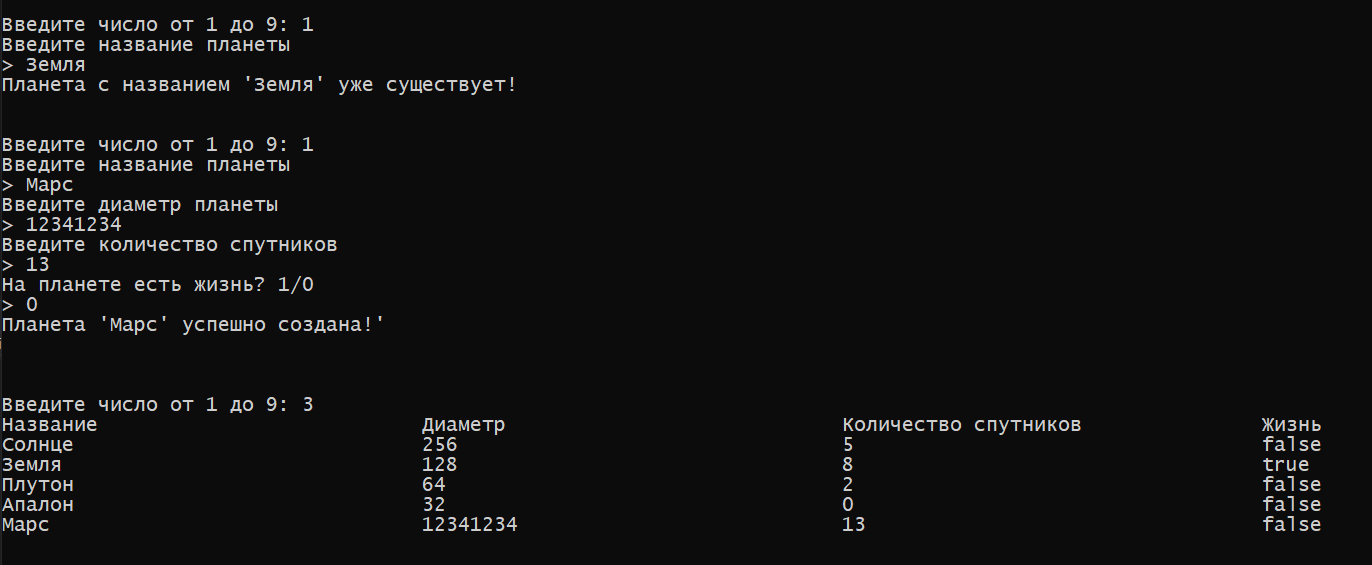
**}**

Анализ результатов

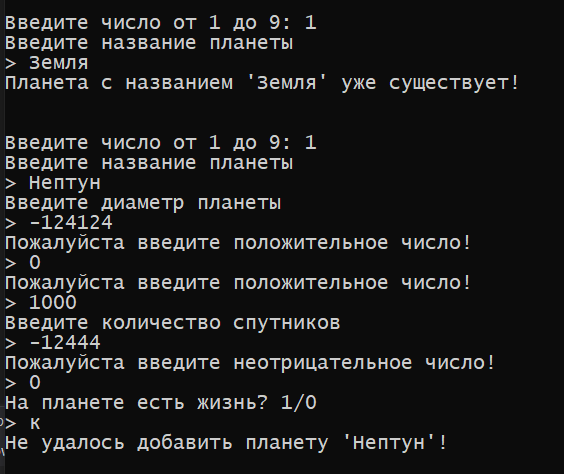
Интерфейс программы



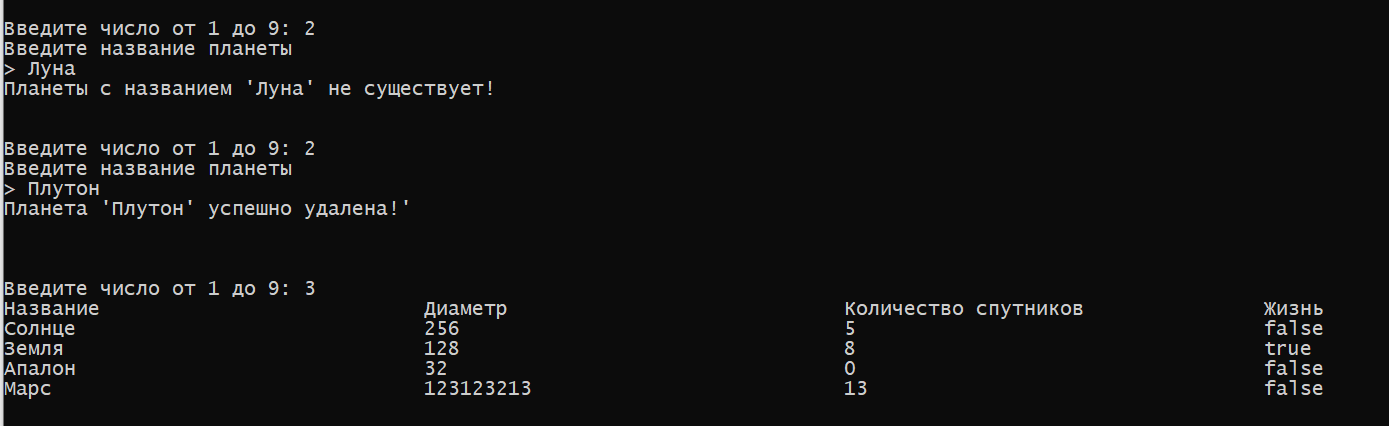
Добавление новой планеты



Проверка на корректность входных данных



Удаление планет



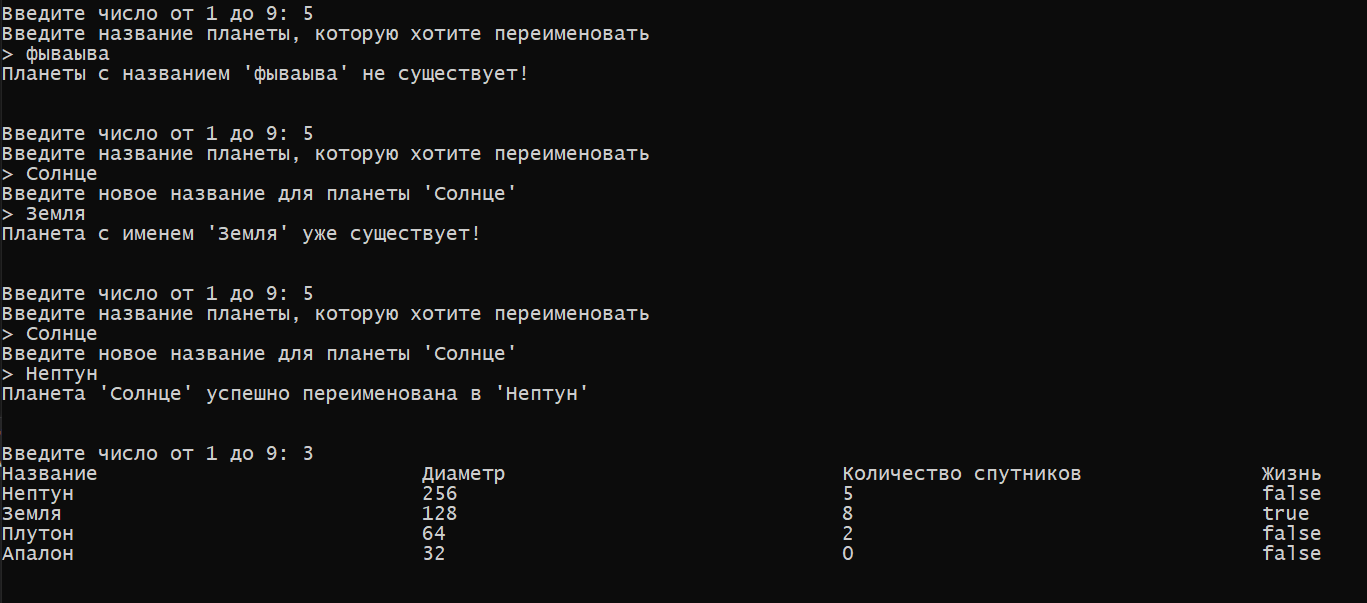
Вывод всех информации о всех планетах



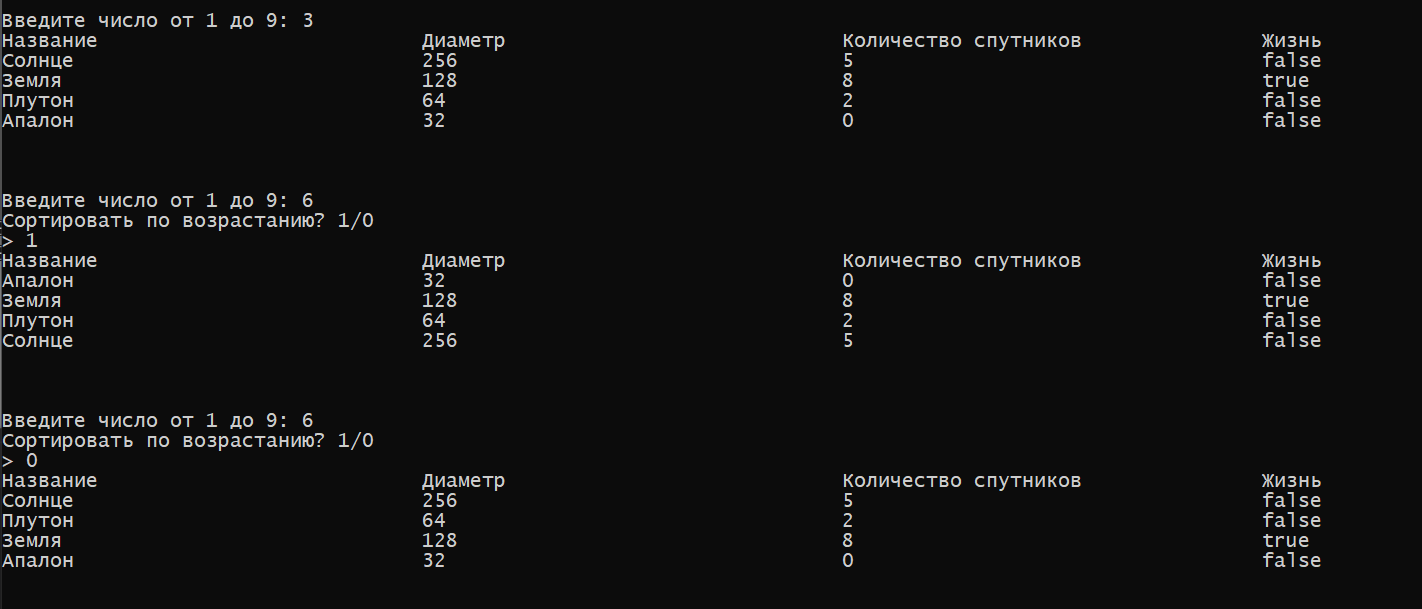
Вывод информации об одной планете



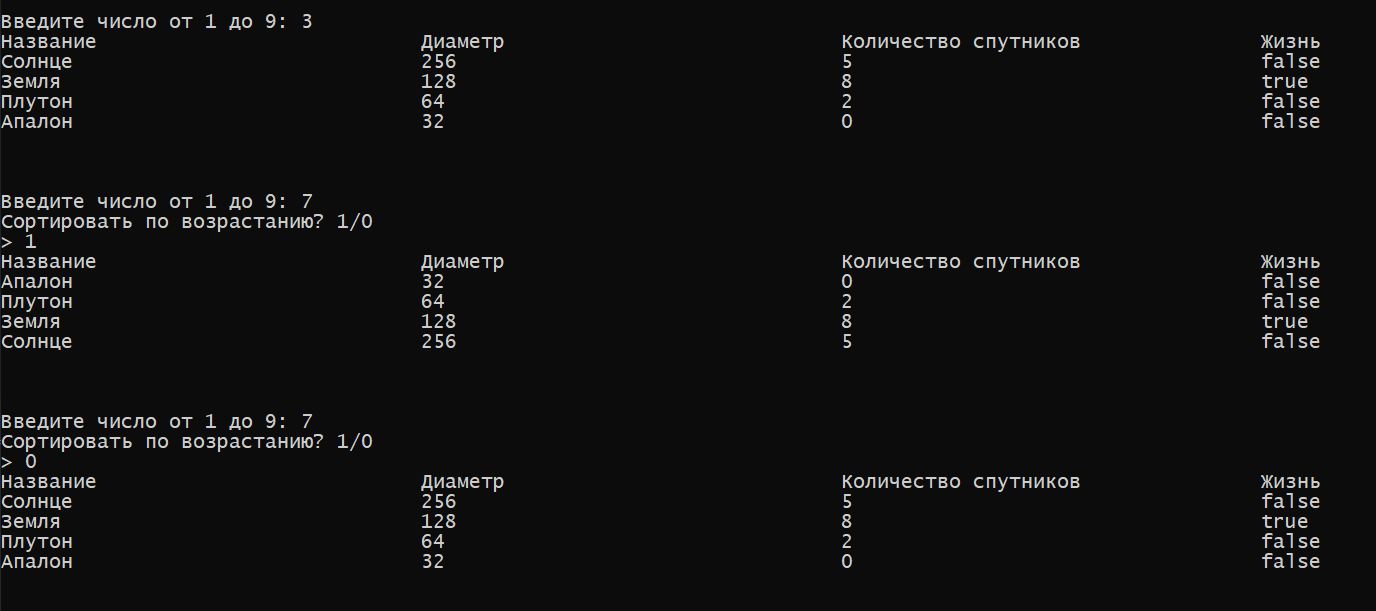
Переименование планеты



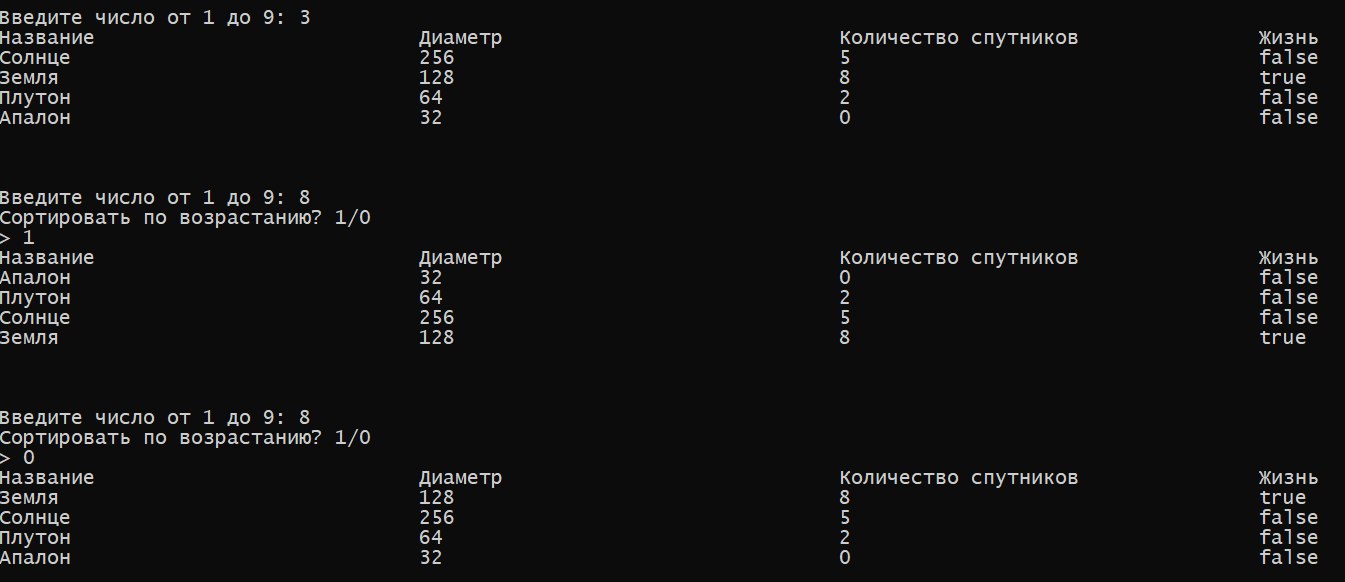
Сортировка планет по названию



Сортировка планет по диаметру



Сортировка планет по количеству спутников



Вывод

Я научился

* Работать со свойствами и методами классов
* Работать с конструкторами и деконструкторами классов
* Работать с экземплярами классов