|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  **Ивангородский гуманитарно-технический институт (филиал)**  **федерального государственного автономного образовательного учреждения высшего образования**  **«Санкт-Петербургский государственный университет аэрокосмического приборостроения»**  **(ИФ ГУАП)** | | | | | | |
|  | | | | | | |
| КАФЕДРА № 2 «Прикладной математики, информатики и информационных таможенных технологий» | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| ОЦЕНКА | | | | | | |
|  | | | | | | |
| ПРЕПОДАВАТЕЛЬ | | | | | | |
|  | | | | | | |
| к.т.н., | | |  |  |  | А.В. Дагаев |
| должность, уч. степень, звание | | |  | подпись, дата |  | инициалы, фамилия |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ№4 | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| Конструирование простейшего класса в С++ | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| по дисциплине:  С++ | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| РАБОТУ ВЫПОЛНИЛ | | | | | | |
|  | | | | | | |
| СТУДЕНТ ГР. № | | 922K |  |  |  | Ю. В. Булыгин |
|  | | номер группы |  | подпись, дата |  | инициалы, фамилия |
|  | | | | | | |
|  | | | | | | |
| Студенческий билет № |  | |  | | | |
|  |  | |  | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| Ивангород 2021 | | | | | | |

**Цель работы:**

Познакомиться с основными понятиями объектно-ориентированного программирования (класс, объект, свойство, метод, конструктор, деструктор), выполнить конструирование класса, предназначенного для хранения заданной структуры данных, изучить способы создания объектов.

**Задание:**

В работе требуется сконструировать класс с заданным набором свойств. Набор свойств следует взять из лабораторной работы № 1 в соответствии со своим вариантом. В класс также должно быть добавлено достаточное количество методов для просмотра и редактирования значений любого из свойств (можно однократное).

Требования к конструированию класса: доступ к свойствам — закрытый, к методам — открытый. В классе следует предусмотреть конструктор по умолчанию, конструктор с параметрами.

**Средаразработки:**

Microsoft Visual Studio 2019



**Описание программы:**

Данная программа предназначена для:

1. Создания объекта с помощью конструктора по умолчанию.
2. Редактирования свойств объекта с помощью конструктора с параметрами.
3. Просмотра свойств объекта.
4. Нахождение самой новой программы.

Для реализации программы было использован тип данных класс.

Класс — это пользовательский тип данных, объединяющий данные и алгоритмы для обработки этих данных. Класс моделирует группу каких-либо реальных объектов (студенты, машины), процессов (путешествия), явлений (погода).

Данные класса представлены в виде переменных и называются свойствами.

Алгоритмы представлены в виде функций и называются методами.

В классе существует разграничение доступа к его членам. Внутреннюю (закрытую) часть класса, доступную только этому классу, составляет раздел private, защищенная часть класса доступна классу и его наследникам — раздел protected, доступны для любых объектов без ограничения члены класса из раздела public (открытая часть класса).

В программе был создан класс:

private: закрытая часть класса

class Prog, содержащий поля:

название программы (nazv)

разработчик (razr)

версия (ver)

год разработки (year)

public: // открытая часть класса

prog() {}// Конструктор по умолчанию создает объект с не инициализированными свойствами

prog (string valNazv, string valRazr, string valVer, int valYear) // Конструктор с параметрами создает объект с заранее определенным набором свойств

void set\_nazv (string valNazv) //Метод, устанавливающий

// новое значение свойству

string get\_nazv() // Метод, позволяющий

//прочитать значение свойства

Логика работа программы:

1.На вход подаются данные о 4х программах: название программы, разработчик, версия, год разработки.

2. Выводится список из 4х строк с названием, разработчиком, версией и годом.

3.У пользователя запрашивают номер строки для редактирования. По введенному номеру последовательно запрашиваются поля: название программы, разработчик, версия, год разработки.

4. Выводится данные о программе имеющей наибольшее значение поля год разработки.

**Пример работы программы:**

На приведённых ниже рисунках представлен пример работы программы.

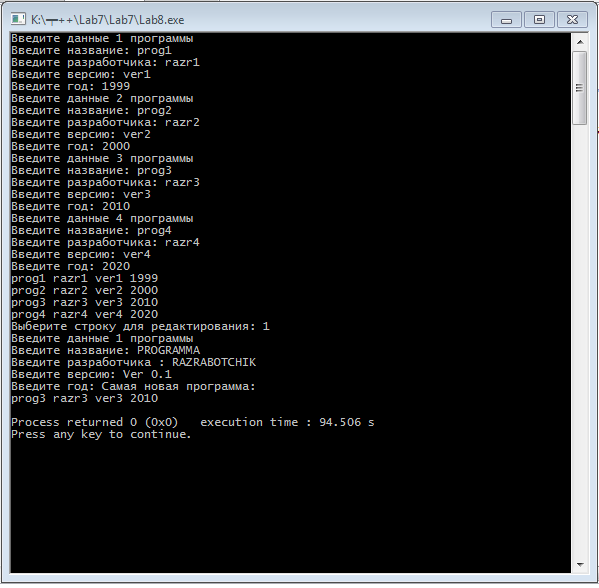


Рисунок 1 – Создание объектов классов с помощью конструктора по умолчанию. Редактирование свойств объекта с помощью конструктора с параметрами и нахождение самой новой программы

**Вывод:**

В результате проделанной работы были получены следующие результаты:

1. Изучено создание и использование классов на основе MVS 2019 для Windows;
2. Разработана программа, выполняющая поиск самой новой программы и предоставляющая возможность просмотра и редактирования свойств объектов классов;
3. Инициализация объектов класса производится с помощью конструкторов по умолчанию, редактирование – с помощью конструкторов с параметрами.

**Код программы**

**#include <iostream>**

**#include <string>**

**#include <windows.h>**

**#include <string>**

**using namespace std;**

**class prog**

**{**

**public:**

**prog() {}//Конструктор по умолчанию создает объект с не инициализированными свойствами**

**prog (string valNazv, string valRazr, string valVer, int valYear) // Конструктор с параметрами создает объект с заранее определенным набором свойств**

**{**

**nazv = valNazv;**

**razr = valRazr;**

**ver = valVer;**

**year = valYear;**

**}**

**void set\_nazv(string valNazv)**

**{**

**nazv = valNazv;**

**}**

**void set\_razr(string valRazr)**

**{**

**razr = valRazr;**

**}**

**void set\_ver(string valVer)**

**{**

**ver = valVer;**

**}**

**void set\_year(int valYear)**

**{**

**year = valYear;**

**}**

**string get\_nazv()**

**{**

**return nazv;**

**}**

**string get\_razr()**

**{**

**return razr;**

**}**

**string get\_ver()**

**{**

**return ver;**

**}**

**int get\_year()**

**{**

**return year;**

**}**

**private :**

**string nazv;**

**string razr;**

**string ver;**

**int year;**

**};**

**int main()**

**{**

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

**prog programm[4];**

**for (int i = 0; i <= 3; i++)**

**{**

**string nazv, razr, ver;**

**int year;**

**cout << "Введите данные " << i + 1 << " программы" << endl;**

**cout << "Введите название: ";**

**cin >> nazv;**

**programm[i].set\_nazv(nazv);**

**cout << "Введите разработчика: ";**

**cin >> razr;**

**programm[i].set\_razr(razr);**

**cout << "Введите версию: ";**

**cin >> ver;**

**programm[i].set\_ver(ver);**

**cout << "Введите год: ";**

**cin >> year;**

**programm[i].set\_year(year);**

**}**

**for (int i = 0; i <= 3; i++)**

**{**

**cout << programm[i].get\_nazv() + " ";**

**cout << programm[i].get\_razr() + " ";**

**cout << programm[i].get\_ver() + " ";**

**cout << programm[i].get\_year() << " " << endl;**

**}**

**cout << "Выберите строку для редактирования: ";**

**int red;**

**cin >> red;**

**for (int i = 0; i < 1; i++)**

**{**

**string nazv, razr, ver;**

**int year;**

**cout << "Введите данные " << red << " программы" << endl;**

**cout << "Введите название: ";**

**cin >> nazv;**

**programm [red - 1].set\_nazv(nazv);**

**cout << "Введите разработчика : ";**

**cin >> razr;**

**programm [red - 1].set\_razr(razr);**

**cout << "Введите версию: ";**

**cin >> ver;**

**programm [red - 1].set\_ver(ver);**

**cout << "Введите год: ";**

**cin >> year;**

**programm [red - 1].set\_year(year);**

**}**

**int last\_prog = programm[0].get\_year();**

**int string = 0;**

**for (int i = 0; i < 3; i++)**

**{**

**if (last\_prog < programm[i].get\_year())**

**{**

**last\_prog = programm[i].get\_year();**

**string = i;**

**}**

**}**

**cout << "Самая новая программа: " << endl;**

**cout << programm[string].get\_nazv() + " ";**

**cout << programm[string].get\_razr() + " ";**

**cout << programm[string].get\_ver() + " ";**

**cout << programm[string].get\_year() << " " << endl;**

**return 0;**

**}**