**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

Дисциплина: «Технология программирования»

Лабораторная работа №6.

Тема: «Конструирование простейшего класса»

Выполнили: Студенты группы РЦТ-22

Балан К. А.

Орехва В. Э.

Принял: к.т.н, доцент кафедры ПИиВТ

Дагаев А. В.

Санкт-Петербург

2023г

**Постановка задачи**

*Вариант 1*

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.

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

Действия, выполняемые программой:

1. создание объекта с помощью конструктора по умолчанию,

2. создание объекта с помощью конструктора с параметрами,

3. создание массива объектов,

4. редактирование и просмотр свойств каждого объекта (можно однократное).

5. обработка массива объектов в соответствии с заданием лабораторной работы 1.

Ввод исходных данных осуществляется с клавиатуры, вывод на экран.

**Описание класса**

class Program

{

public:

Program();

Program(std::string n, std::string d, std::string v, int y);

~Program();

std::string getName();

void setName(std::string val);

std::string getDeveloper();

void setDeveloper(std::string val);

std::string getVersion();

void setVersion(std::string val);

int getYear();

void setYear(int val);

private:

std::string name;

std::string developer;

std::string version;

int year;

};

Класс Program состоит из:

1). Закрытых членов класса:

name (string) - название программы.

developer (string) - разработчик программы.

version (string) - версия программы.

year (int) - год выпуска программы.

2). Публичных методов:

Program() - конструктор по умолчанию, который инициализирует программу с пустыми значениями.

Program(string n, string d, string v, int y) - конструктор, позволяющий установить значения для программы при создании объекта.

string getName() - метод для получения названия программы.

void setName(string val) - метод для установки значения названия программы.

string getDeveloper() - метод для получения разработчика программы.

void setDeveloper(string val) - метод для установки значения разработчика программы.

string getVersion() - метод для получения версии программы.

void setVersion(string val) - метод для установки значения версии программы.

int getYear() - метод для получения года выпуска программы.

void setYear(int val) - метод для установки значения года выпуска программы.

3). Использование класса в функции main():

Создание объекта program1 с использованием конструктора по умолчанию и объекта program2 с использованием конструктора с параметрами.

Установка значений для объекта program1 с использованием методов set.

Редактирование значений объекта program2 с использованием методов set.

Получение и отображение информации о программе, представленной объектом program2, с использованием методов get.

Этот класс позволяет представлять информацию о программе, устанавливать и изменять данные и получать информацию о программе для дальнейшего использования.

**Описание программы**

В начале программы подключаются необходимые библиотеки - iostream для ввода-вывода и string для работы со строками. Затем объявляется класс Program с приватными свойствами name (название программы), developer (разработчик), version (версия) и year (год выпуска).

В функции main() создается массив объектов класса Program под названием Prog. Затем с помощью цикла пользователю предлагается ввести данные для каждой программы (название, разработчик, версию и год выпуска), и значения записываются в объекты массива Prog с помощью соответствующих методов set. После завершения цикла, для каждой программы из массива Prog выводится название и год выпуска с помощью методов get. Программа заканчивается с помощью return 0.

**Описание методов**

В коде определены методы класса Program и основная функция main. Рассмотрим каждый метод и их назначение:

Program() - конструктор класса Program без параметров. Инициализирует объект класса с значениями по умолчанию для каждого поля (Noname, NoDeveloper, NoVersion, 0).

Program(std::string n, std::string d, std::string v, int y) - конструктор класса Program с параметрами. Инициализирует объект класса с заданными значениями для полей name, developer, version и year.

~Program() - деструктор класса Program. Не выполняет никаких действий, так как необходимость в явном вызове деструктора отсутствует.

std::string getName() - метод получения значения поля name объекта класса Program.

void setName(std::string val) - метод установки значения для поля name объекта класса Program.

std::string getDeveloper() - метод получения значения поля developer объекта класса Program.

void setDeveloper(std::string val) - метод установки значения для поля developer объекта класса Program.

std::string getVersion() - метод получения значения поля version объекта класса Program.

void setVersion(std::string val) - метод установки значения для поля version объекта класса Program.

int getYear() - метод получения значения поля year объекта класса Program.

void setYear(int val) - метод установки значения для поля year объекта класса Program.

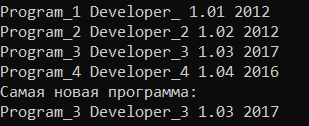
В функции main создается массив объектов класса Program (Prog), затем с помощью цикла пользователю предлагается ввести данные для каждой программы, и значения записываются в объекты массива Prog с помощью соответствующих методов set. После завершения цикла, для каждой программы из массива Prog выводится название и год выпуска с помощью методов get.

**Введенные данные**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Разработчик** | **Версия** | **Год выпуска** |
| Program\_1 | Developer\_ | 1.01 | 2012 |
| Program\_2 | Developer\_2 | 1.02 | 2012 |
| Program\_3 | Developer\_3 | 1.03 | 2017 |
| Program\_4 | Developer\_4 | 1.04 | 2016 |

**Результат работы**

Программа выводит данные о программах, а так же находит и выводит на экран самую новую программу (Рис. 1). В первом столбце написано название программы, во втором имя разработчика, в третьем версия и в четвертом год выпуска программы.

****Рис. 1

**Вывод**

В ходе проделанной работы были получены следующие результаты:

1) Разработали программу, использующую пользовательский тип данных класс и его свойства, такие как разграничение доступа к его членам (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).

2) Написали код, используя специальный метод - конструктор.

**Листинг**

#include <iostream>

using namespace std;

// Объявление класса

class Program

{

public:

Program();

Program(std::string n, std::string d, std::string v, int y);

~Program();

std::string getName();

void setName(std::string val);

std::string getDeveloper();

void setDeveloper(std::string val);

std::string getVersion();

void setVersion(std::string val);

int getYear();

void setYear(int val);

private:

std::string name;

std::string developer;

std::string version;

int year;

};

// Программа, использующая класс

int main()

{

system("chcp 1251");

Program Prog[4];

std::string name;

std::string developer;

std::string version;

int year;

int max\_year = 0;

int max\_array = 0;

for (int i = 0; i < 4; i++)

{

cout << "Введите название " << i + 1 << " программы: ";

cin >> name;

Prog[i].setName(name);

cout << "Введите разработчика " << i + 1 << " программы: ";

cin >> developer;

Prog[i].setDeveloper(developer);

cout << "Введите версию " << i + 1 << " программы: ";

cin >> version;

Prog[i].setVersion(version);

cout << "Введите год выпуска " << i + 1 << " программы: ";

cin >> year;

Prog[i].setYear(year);

if (max\_year < year) {

max\_year = year;

max\_array = i;

}

}

for (int i = 0; i < 4; i++)

cout << Prog[i].getName() << " " << Prog[i].getDeveloper() << " " << Prog[i].getVersion() << " " << Prog[i].getYear() << endl;

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

cout << Prog[max\_array].getName() << " " << Prog[max\_array].getDeveloper() << " " << Prog[max\_array].getVersion() << " " << Prog[max\_array].getYear() << endl;

return 0;

}

// Реализация функций-членов класса

Program::Program()

{

// Конструктор по умолчанию

name = "Noname";

developer = "NoDeveloper";

version = "NoVersion";

year = 0;

}

Program::Program(std::string n, std::string d, std::string v, int y)

{

// Конструктор с параметрами

name = n;

developer = d;

version = v;

year = y;

}

Program::~Program()

{

// Деструктор по умолчанию, можно не включать в класс,

// явно в программе не вызывается

}

std::string Program::getName()

{

return name; // Возвращает значение свойства name

}

void Program::setName(std::string val)

{

name = val; // Перезаписывает значение свойства name

}

std::string Program::getDeveloper()

{

return developer; // Возвращает значение свойства developer

}

void Program::setDeveloper(std::string val)

{

developer = val; // Перезаписывает значение свойства developer

}

std::string Program::getVersion()

{

return version; // Возвращает значение свойства version

}

void Program::setVersion(std::string val)

{

version = val; // Перезаписывает значение свойства version

}

int Program::getYear()

{

return year; // Возвращает значение свойства year

}

void Program::setYear(int val)

{

year = val; // Перезаписывает значение свойства year

}