|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 №5 | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| Конструирование класса моделирующего работу устройства | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| 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 понятием инкапсуляция в объектно­ориентированном программировании, научиться защищать свойства объекта, моделировать поведение предмета.

**Задание**

Разработать модель работы устройства, состоящую из 3-­4 свойства, наиболее существенных для описания поведения устройства, и 3-­6 методов, моделирующих поведение объекта.

Определить набор допустимых значений для каждого свойства.

На основе модели сконструировать класс. Доступ к свойствам класса должен быть закрытым (private), к методам – открытым (public); изменение значений свойств (состояния) объекта осуществляется соответствующими методами.

Разработать консольное приложение, демонстрирующее работу объекта, в составе: главный модуль, содержащий функцию main,

* + модуль, содержащий разработанный класс.

Программа должна выполнять следующие действия:

* + создание объекта;
  + демонстрация текущего состояния объекта с помощью сообщений на экране;



**Среда разработки**

Microsoft Visual Studio 2019

**Описание программы**

Описание полей и методов класса:

private: закрытая часть класса

class Svetofor, содержащий поля:

поле cvet - цвет светофора

поле pesh - отсек светофора для пешеходов

поле strelka – отсек светофора отвечающий за поворот машин

public: открытая часть класса:

string get\_cvet() // метод получения состояния поля цвета

void set\_cvet(string c) // метод установки поля цвета

string get\_strelka()// метод получения состояния поля стрелка

void set\_strelka(string s) // метод установки поля стрелка

string get\_peshexod() // метод получения состояния поля пешеход

void set\_peshexod(string p) // метод установки поля пешеход

void Timer() // Метод организующий время работы состояний светофора

**Логика работа программы**

Создаётся объект класса Svetofor. Запускается бесконечный цикл. Посредством методов set происходит установка состояния полей светофора. Посредством методов get получается и выводится через cout состояние полей – цвет, стрелка, пешеход. Посредством метода timer производится ожидание до переключения на следующее состояние полей и выводится время ожидания до переключения следующего состояния.

**Результат работы программы**

На приведённых ниже рисунках представлен пример работы программы.

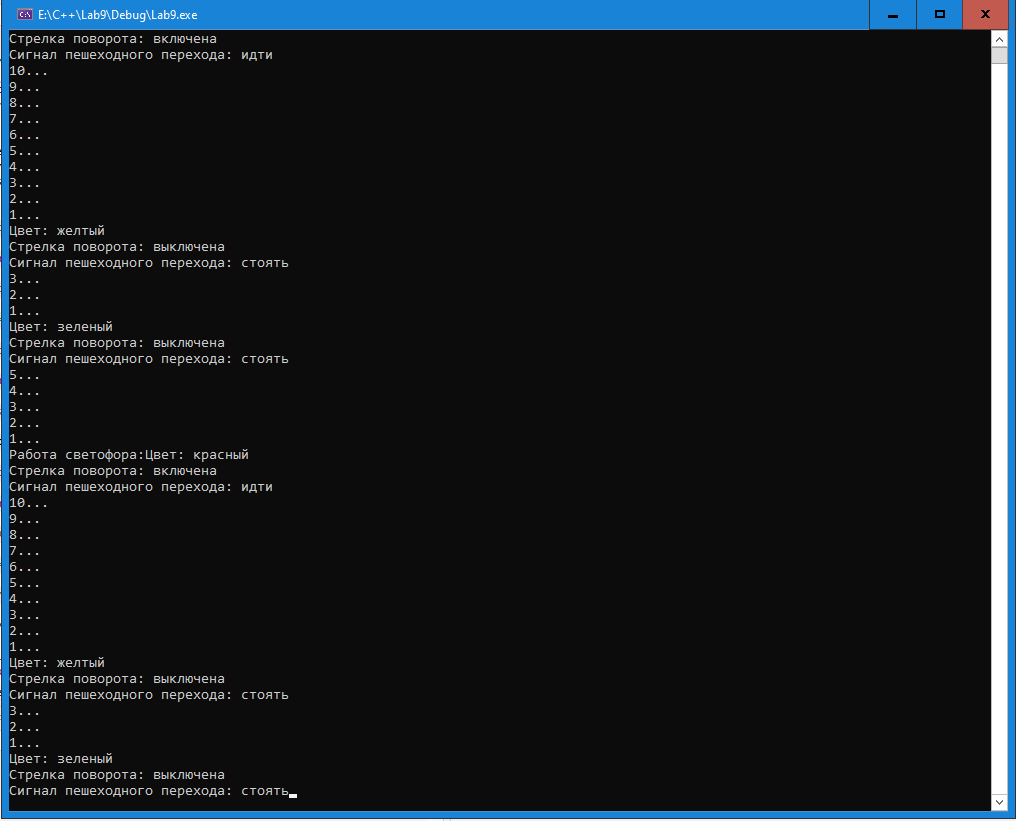


Рисунок 1 –результат работы программы

**Выводы**

В результате проделанной работы были получены следующие результаты:

1. Произведено ознакомление с понятием инкапсуляция в объектно­ориентированном программировании, научиться защищать свойства объекта, моделировать поведение предмета.
2. Разработана модель работы устройства, состоящую из 3-­4 свойства, наиболее существенных для описания поведения устройства, и 3-­6 методов, моделирующих поведение объекта.
3. Разработано консольное приложение, демонстрирующее работу объекта, в составе: главный модуль, содержащий функцию main,
   1. модуль, содержащий разработанный класс.

Программа должна выполнять следующие действия:

* 1. создание объекта;
  2. демонстрация текущего состояния объекта с помощью сообщений на экране;

**Код программы**

#include <iostream>

#include <string>

#include <windows.h>

using namespace std;

class Svetofor

{

private:

string cvet = "красный";

string strelka = "включена";

string pesh = "идти";

public:

string get\_cvet()

{

return cvet;

}

void set\_cvet(string c)

{

cvet = c;

}

string get\_strelka()

{

return strelka;

}

void set\_strelka(string s)

{

strelka = s;

}

string get\_peshexod()

{

return pesh;

}

void set\_peshexod(string p)

{

pesh = p;

}

void Timer()

{

if (cvet == "красный")

{

Sleep(1000);

for (int i=10; i > 0; i--)

{

cout << "\n" << i << "...";

Sleep(1000);

}

}

if (cvet == "желтый")

{

Sleep(1000);

for (int i = 3; i > 0; i--)

{

cout << "\n" << i << "...";

Sleep(1000);

}

}

if (cvet == "зеленый")

{

Sleep(1000);

for (int i = 5; i > 0; i--)

{

cout << "\n" << i << "...";

Sleep(1000);

}

}

}

};

int main()

{

setlocale(LC\_ALL, "rus");

Svetofor Svet;

while (1 < 2)

{

cout << "Работа светофора:";

cout << "Цвет: " << Svet.get\_cvet() << "\n" << "Cтрелка поворота: " << Svet.get\_strelka() << "\n" << "Сигнал пешеходного перехода: " << Svet.get\_peshexod();

Svet.Timer();

string c = "желтый";

Svet.set\_cvet(c);

string s = "выключена";

Svet.set\_strelka(s);

string p = "стоять";

Svet.set\_peshexod(p);

cout << "\n";

cout << "Цвет: " << Svet.get\_cvet() << "\n" << "Cтрелка поворота: " << Svet.get\_strelka() << "\n" << "Сигнал пешеходного перехода: " << Svet.get\_peshexod();

Svet.Timer();

c = "зеленый";

Svet.set\_cvet(c);

s = "выключена";

Svet.set\_strelka(s);

p = "стоять";

Svet.set\_peshexod(p);

cout << "\n";

cout << "Цвет: " << Svet.get\_cvet() << "\n" << "Cтрелка поворота: " << Svet.get\_strelka() << "\n" << "Сигнал пешеходного перехода: " << Svet.get\_peshexod();

Svet.Timer();

c = "красный";

Svet.set\_cvet(c);

s = "включена";

Svet.set\_strelka(s);

p = "идти";

Svet.set\_peshexod(p);

cout << "\n";

}

}