**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 №7.

Тема: «Конструирование класса, моделирующего работу устройства»

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)* Дорожный светофор

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.

**Описание класса**

class TrafficLight {

public:

enum class State {

Normal,

Emergency,

Off

};

State get() const {}

void simulateLight(int numCycles, double duration) {}

void simulateLightYellow(int numCycles, double duration) {}

void displayState(const TrafficLight::State& state) {}

void setNormalMode() {}

void setEmergencyMode() {}

void turnOff() {}

};

Класс TrafficLight:

Данный класс предназначен для указания режимов работы светофора.

Открытый доступ:

Определяем перечисление под названием State, которое определяет возможные состояния светофора.

Перечисление State имеет три имени: Normal, Emergency и Off, которые представляют нормальное состояние светофора, аварийное состояние светофора и выключенное состояние светофора соответственно.

Метод simulateLight имитирует работу светофора в обычном режиме и принимает два параметра: numCycles - количество циклов переключения светофора, и duration - длительность каждой фазы светофора.

Метод simulateLight\_yellow имитирует работу светофора в аварийном режиме, когда горит только желтый мигающий сигнал. Она также принимает два параметра: numCycles - количество циклов переключения светофора, и duration - длительность каждой фазы светофора.

Метод displayState отображает текущее состояние светофора в зависимости от переданного значения state, которое является экземпляром перечисления State.

Метод setNormalMode устанавливает состояние светофора в обычный режим, изменяя значение state на TrafficLight::State::Normal. Затем она вызывает функцию simulateLight для имитации работы светофора и выводит текущее состояние.

Метод setEmergencyMode устанавливает состояние светофора в аварийный режим, изменяя значение state на TrafficLight::State::Emergency. Затем она вызывает функцию simulateLight\_yellow для имитации работы светофора и выводит текущее состояние.

Метод turnOff выключает светофор, изменяя значение state на TrafficLight::State::Off. Затем она выводит текущее состояние.

**Описание программы**

В этом коде создается класс TrafficLight с приватным перечислением State, содержащим состояния светофора: Normal (обычный режим), Emergency (аварийный режим) и Off (выключено). Класс имеет публичные методы для отображения текущего состояния светофора, переключения светофора в обычный и аварийный режимы, а также выключения светофора.

В функции main() создается объект класса TrafficLight с начальным состоянием в обычном режиме. Затем в цикле while пользователю предлагается выбрать действие: переключить светофор в обычный режим, аварийный режим, выключить светофор или отобразить текущее состояние. В зависимости от выбора пользователя вызываются соответствующие методы объекта trafficLight. После выполнения выбранного действия вызывается метод startSimulation(), который моделирует работу светофора в соответствующем режиме.

Этот код демонстрирует использование класса TrafficLight для управления светофором через консольное взаимодействие.

**Результат работы программы**

На Рис.1 представлена консоль, в которой отображается меню, реализующее управление светофором. В нем доступно три режима, отображение текущего состояния и выход из программы.

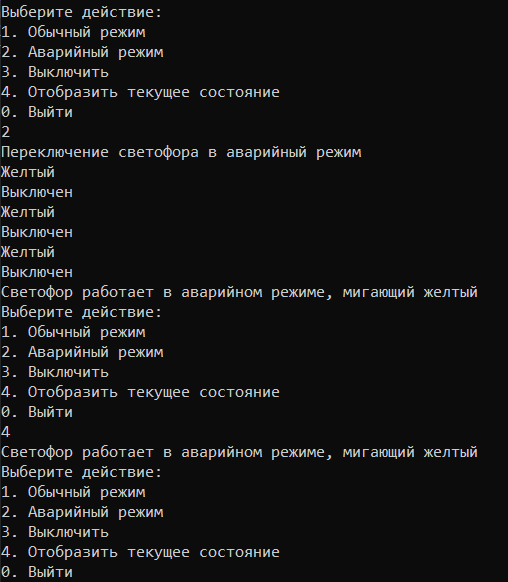
****

Рис.1 Пример работы программы

**Вывод**

В ходе проделанной работы были получены следующие результаты:

1. Написали код, в котором создали объект класса TrafficLight с начальными значениями и использовали открытый метод для управления состоянием светофора.

**Листинг**

#include <iostream>

#include <thread>

#include <chrono>

class TrafficLight {

public:

enum class State {

Normal,

Emergency,

Off

};

};

void simulateLight(int numCycles, double duration) {

for (int i = 0; i < numCycles; ++i) {

std::cout << "Красный" << std::endl;

std::this\_thread::sleep\_for(std::chrono::seconds(static\_cast<int>(duration)));

std::cout << "Желтый" << std::endl;

std::this\_thread::sleep\_for(std::chrono::seconds(static\_cast<int>(duration)));

std::cout << "Зеленый" << std::endl;

std::this\_thread::sleep\_for(std::chrono::seconds(static\_cast<int>(duration)));

}

}

void simulateLight\_yellow(int numCycles, double duration) {

for (int i = 0; i < numCycles; ++i) {

std::cout << "Желтый" << std::endl;

std::this\_thread::sleep\_for(std::chrono::seconds(static\_cast<int>(duration)));

std::cout << "Выключен" << std::endl;

std::this\_thread::sleep\_for(std::chrono::seconds(static\_cast<int>(duration)));

}

}

void displayState(const TrafficLight::State& state) {

switch (state) {

case TrafficLight::State::Normal:

std::cout << "Светофор работает в обычном режиме" << std::endl;

break;

case TrafficLight::State::Emergency:

std::cout << "Светофор работает в аварийном режиме, мигающий желтый" << std::endl;

break;

case TrafficLight::State::Off:

std::cout << "Светофор выключен" << std::endl;

break;

}

}

void setNormalMode(TrafficLight::State& state) {

state = TrafficLight::State::Normal;

std::cout << "Переключение светофора в обычный режим" << std::endl;

simulateLight(3, 1);

displayState(state);

}

void setEmergencyMode(TrafficLight::State& state) {

state = TrafficLight::State::Emergency;

std::cout << "Переключение светофора в аварийный режим" << std::endl;

simulateLight\_yellow(3, 1);

displayState(state);

}

void turnOff(TrafficLight::State& state) {

state = TrafficLight::State::Off;

std::cout << "Выключение светофора" << std::endl;

displayState(state);

}

int main() {

TrafficLight::State state = TrafficLight::State::Normal;

int work = 1, command;

system("chcp 1251");

while (work) {

std::cout << "Выберите действие:" << std::endl;

std::cout << "1. Обычный режим" << std::endl;

std::cout << "2. Аварийный режим" << std::endl;

std::cout << "3. Выключить" << std::endl;

std::cout << "4. Отобразить текущее состояние" << std::endl;

std::cout << "0. Выйти" << std::endl;

std::cin >> command;

switch (command) {

case 1:

setNormalMode(state);

break;

case 2:

setEmergencyMode(state);

break;

case 3:

turnOff(state);

break;

case 4:

displayState(state);

break;

case 0:

work = 0;

break;

default:

std::cout << "Неверный выбор. Пожалуйста, попробуйте ещё раз." << std::endl;

}

}

return 0;

}