Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)

Факультет «Робототехника и комплексная автоматизация»

Кафедра «Системы автоматизированного проектирования»

**Отчет по домашнему заданию**

По курсу «Объектно-ориентированное программирование»

Выполнил: Студент Сергеева Д.К.

Группа РК6-26Б

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2020 г.

**ООП Вариант Е20**

**Задание:**

Разработать объектно-ориентированную программу угловой закраски алфавитно-цифрового окна консоли косыми фронтами его знакомест справа налево и снизу-вверх. Такая закраска должна выполняться поочередно во все цвета фона в порядке убывания их escape-кодов. Чередование фонов должно продолжаться до завершения программы при любом консольном вводе или по сигналу ^C. При этом должен быть восстановлен исходный фон с очисткой окна консоли. В программе должны быть реализованы необходимые манипуляторы потока стандартного вывода для строк escape-последовательностей. Их программный код вместе с классом escape-потока, перегрузкой оператора вывода его объектов и функцией контроля консольного ввода нужно сосредоточить в консольном пространстве имен. Цикл перекраски должна кодировать основная функция программы.

**Алгоритм:**

Для работы программы пишем своё пространство имен, в котором находится функции для закрашивания терминала в определённый цвет с помощью escape последовательности.

В основной функции проходимся по всем позициям терминала с помощью 2 циклов for от и закрашиваем определенную ячейку в соответствующий цвет.

**Входные данные:**

Отсутствуют.

**Выходные данные:**

Любой введенный символ терминал означает прекращение работы программы.

**Текст программы:**

#include <sys/ioctl.h>

#include <iomanip>

#include <string>

#include <sstream>

#include <iostream>

#include <termios.h>

#include <unistd.h>

#include <signal.h>

#include <time.h>

using namespace std;

static int done = 0;

namespace con

{

// Max number of columns in terminal

int maxCol()

{

struct winsize w;

ioctl(0, TIOCGWINSZ, &w);

return (w.ws\_col);

}

// Max number of row in terminal

int maxRow()

{

struct winsize w;

ioctl(0, TIOCGWINSZ, &w);

return (w.ws\_row);

}

// Display cleaning

ostream& DC(ostream& s)

{

return s << string("\033[2J\033[1;1H");

}

// Line cleaning

ostream& LC(ostream& s)

{

return s << string("\033[2K");

}

class escapeStream

{

private:

string escape;

public:

escapeStream(string str) : escape(str) {};

friend ostream& operator<<(ostream &s, escapeStream esc); // Friend overload <<

};

// Friend overload <<

ostream& operator<<(ostream& s, escapeStream esc)

{

s << esc.escape << flush;

return s;

}

// Remove cursor in position

escapeStream curPos(int y, int x)

{

ostringstream s\_out;

s\_out << "\033[" << y << ";" << x << "H";

return escapeStream(s\_out.str());

}

// Set graphic escape manipulator

escapeStream SGM(int num)

{

ostringstream s\_out;

s\_out << "\033[" << num << "m";

return escapeStream(s\_out.str());

}

}

void interruptor(int signo)

{

done = signo;

}

// Check typed symbol in the terminal

int hitKeyBreak()

{

int n = 0;

char buf[512];

struct termios t[2];

tcgetattr(0, &t[0]);

tcgetattr(0, &t[1]);

t[0].c\_lflag &= ~(ICANON | ECHO);

t[0].c\_cc[VMIN] = 0;

t[0].c\_cc[VTIME] = 0;

tcsetattr(0, TCSANOW, &t[0]);

n = read(0, buf, 512);

tcsetattr(0, TCSAFLUSH, &t[1]);

return n;

}

int main(int argc, char\* argv[])

{

int maxrow = con::maxRow(), maxcol = con::maxCol();

cout << con::curPos(maxrow, 1);

cout << con::DC;

cout << con::curPos(maxrow, 1);

cout << maxrow << "x" << maxcol << flush;

maxrow -= 1;

signal(SIGINT, interruptor);

int startColor = 0;

while (done < 1)

{

int color = startColor;

for (int x = maxrow; x > 0; x--)

{

color %= 8;

int k = color;

for (int y = 0; y < maxcol; y++)

{

k %= 8;

cout << con::curPos(x, y) << " ";

cout << con::SGM(47 - k);

k++;

}

++color;

}

++startColor;

startColor %= 8;

cout << con::curPos(maxcol, 10);

if (hitKeyBreak() > 0)

break;

sleep(2);

}

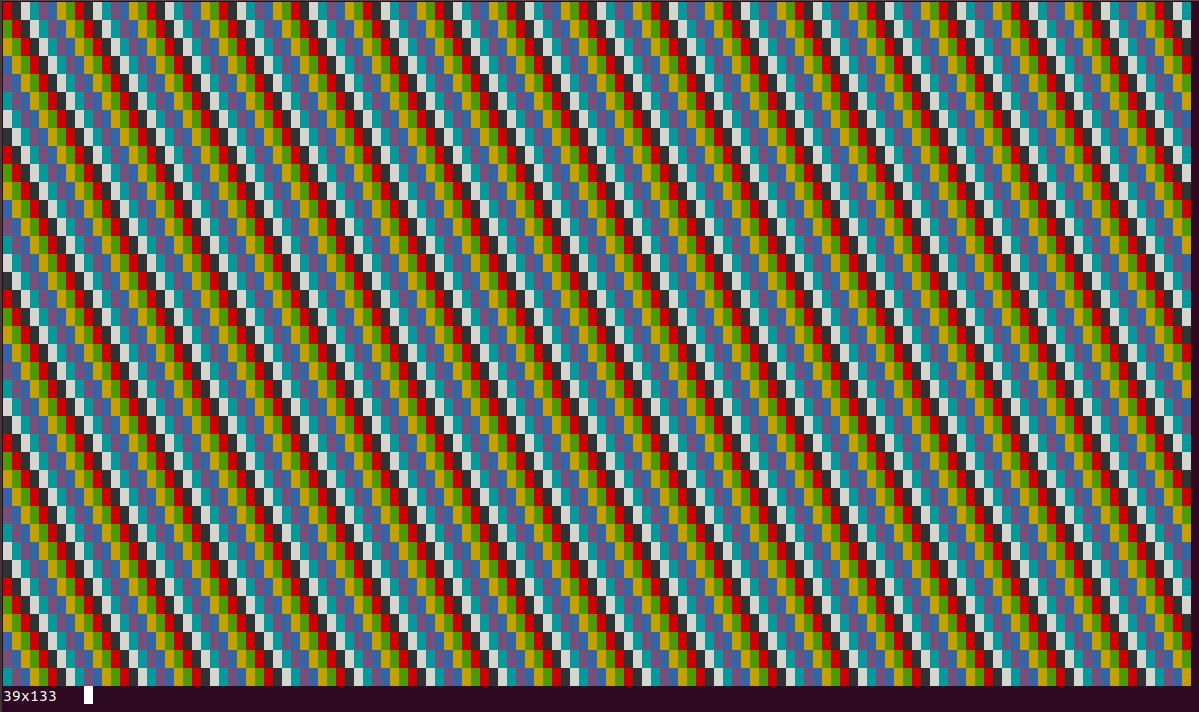
cout << con::SGM(0);

cout << con::DC;

return 0;

}

**Тесты:**

****

**Список использованной литературы:**

* Волосатова Т.М., Родионов С.В. Лекции по курсу «Объектно-ориентированное программирование»
* bigor.bmstu.ru