|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехника и комплексная автоматизация (РК)

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИКИ**

Студент Камалов Антон Павлович

Группа РК6-35Б

Модуль 1

Вариант G16

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Камалов А.П.**

*подпись, дата фамилия, и.о.*

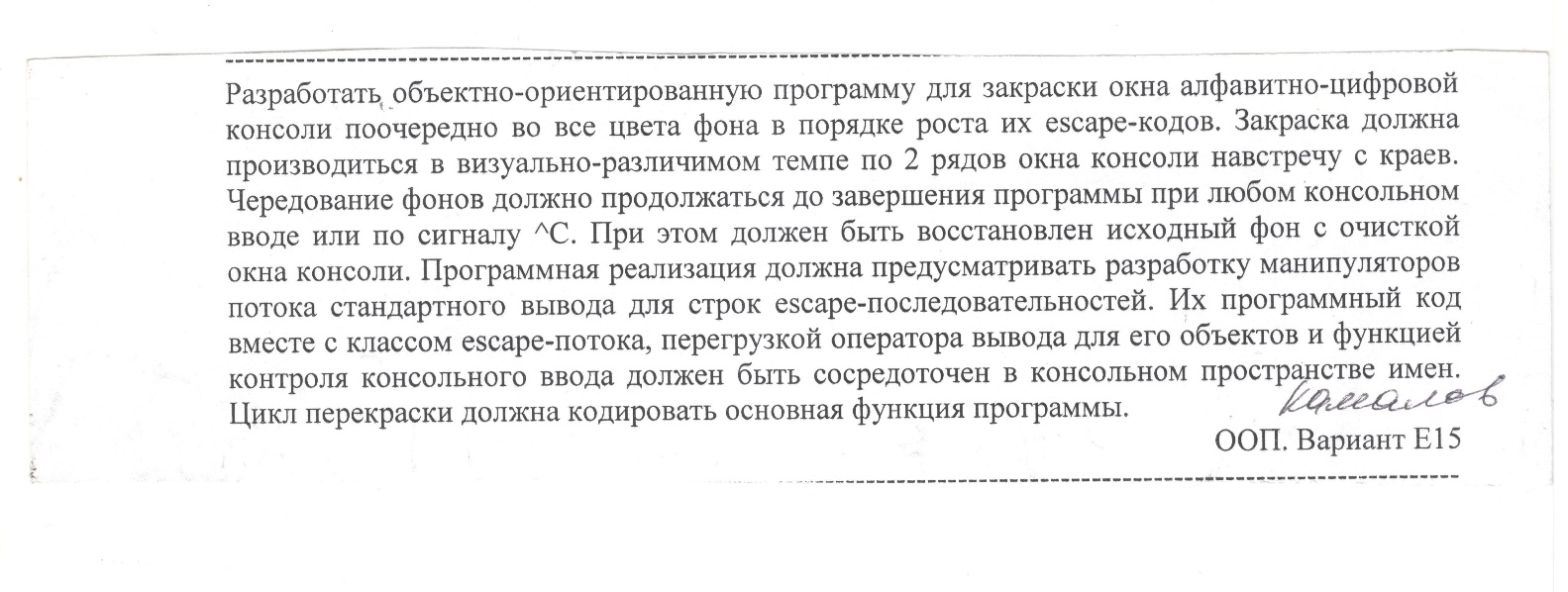
Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузьмина И.А.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2023 г.*

**Задание**

****

**Описание программы**

**Входные данные:** отсутствуют в данной программе.

**Выходные данные:** расположение фигуры на шахматной доске.

**Класс Figure**

*Информационные поля:*

pos (char []) – координаты фигуры на шахматной доске;

cur\_player (int) – переменная, отслеживающая текущего игрока (1 – для игрока, 2 – для компьютера);

*Конструкторы и деструкторы:*

Figure () – конструктор по умолчанию;

Figure (string) – конструктор инициализации;

*Методы:*

operator== (string) – публичный метод перегрузки оператора равенства;

attack (string) – публичный метод атаки фигуры;

isa () – публичный метод маркировки фигуры;

desk () – публичный метод отображения шахматной доски;

deskout (string) – публичный статический метод проверки положения фигуры на доске;

game () – публичный метод запуска игры;

makeMove (string) – публичный метод обновления позиции фигуры на доске.

**Описание алгоритма:**

Для обеспечения работы с изменением цвета фона терминала будем работать с escape-последовательностью (набором символов, который выталкивается в поток для его форматирования). Написание программы начнем с написания манипуляторов. Для написания манипулятора цвета, перемещения курсора необходимо создадим класс estream с приватным строковым полем escape, так как все они имеют параметры. Конструктор класса будет инициализировать это поле строкой, переданной аргументом. Для этого класса будет перегружен оператор вывода. Он будет направит строку в поток. Манипулятор flush будет выталкивать содержимое буфера в поток. Для escape-последовательности манипулятор необходимо явно прописать. Манипулятор цвета принимает в качестве аргумента код цвета, который имеет вид – 40+x, где х ∈ [0,7]. Манипуляторный код сосредоточен в пространстве имён con, куда также входят функции comax() и romax(), хранящие информацию о размерности терминала. Функция kbin() обеспечивает проверку нажатия клавиши ENTER. В буфер заносятся полученные из потока стандартного ввода символы. В переменную flags занесется результат вызова функции fcntrl(0, F\_GETFL). Здесь 0 означает, что действие идет с потоком стандартного ввода. Результат будет использован в функции read, которая будет просматривать буфер до тех пор, пока не будет нажата клавиша ENTER. Kbin() вернет значение большее единицы, когда будет нажата данная клавиша, и после программа завершит свою работу.

Необходимо раскрасить экран терминала по 2 ряда навстречу с краев во все цвета фона в порядке роста их escape-кодов (черный-красный-зеленый-желтый-синий-пурпурный-голубой-белый). Закраска продолжается до ввода с консоли, либо сигнала ^C. Цикл перекраски содержится в основной функции. С помощью функций определения ширины и длины экрана, а также манипуляторов во вложенных циклах меняется положение курсора и цвет, которым красить: курсор пробегает с каждого края по 2 строки слева-направо по каждой строчке, а затем меняет цвет и вновь меняет положение на строчку ниже/выше. Если пользователь изменяет размер терминала, то выполняется функция resize(), которая задаёт новое положение середины окна терминала.

**Список литературы**

1. Пол Айра. Объектно-ориентированное программирование на C++: пер. с англ. 2-е изд. СПБ.: Невский Диалект; М.: Издательство БИНОМ, 2001. 462 с. [Pohl Ira. Object-Oriented Programming Using C++. 2nd ed. Addison-Wesley, 1996. 576 p.].
2. Страуструп Б. Язык программирования C++: пер. с англ / под ред. Н.Н. Мартынова. Специальное изд. М.: Бином, 2011. 1035 c. [Stroustrup B. The C++ Programming Language. Special ed. Addison-Wesley, 2000. 1029p.].
3. Шилдт Г. Самоучитель C++. 3-е изд. СПБ.: БХВ-Петербург, 2002. 688 с. [Schildt H. Teach Yourself C++. 3d ed. McGraw-Hill, 1998. 768 p.].
4. Волосатова Т.М., Родионов С.В. Объектно-ориентированное программирование на С++. Режим доступа: http://bigor.bmstu.ru/?cnt/?doc=VU/base.cou (дата обращения 23.05.2023).

**Приложение 1**

Текст программы

#include<iostream>

#include<sstream>

#include<sys/ioctl.h>

#include<fcntl.h>

#include<signal.h>

#include<unistd.h>

static int done = 0;

static int mid = 0;

using namespace std;

namespace con{

int comax(){

struct winsize w;

ioctl(0,TIOCGWINSZ, &w);

return(w.ws\_col);

}

int romax(){

struct winsize w;

ioctl(0,TIOCGWINSZ, &w);

return(w.ws\_row);

}

class estream{

private:

string escape;

public:

estream(string e) : escape(e) {};

friend ostream& operator<<(ostream&,estream);

};

ostream& operator <<(ostream& s,estream e){

s << e.escape << flush;

return s;

}

ostream& ED(ostream& s){

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

}

estream CUP(int y,int x){

ostringstream sout;

sout << "\033[" << y << ";" << x << "H";

return estream(sout.str());

}

estream SGR(int r){

ostringstream sout;

sout<<"\033["<< r <<"m";

return estream(sout.str());

}

}

using con::SGR;

using con::CUP;

void interruptor(int signo){

done=signo;

return;

}

void resize(int signo) {

if (con::romax() % 2 == 0)

mid = con::romax() / 2;

else

mid = (con::romax() / 2) + 1;

}

int kbin(){

char buf[512];

int n=0;

int flags=fcntl(0,F\_GETFL);

fcntl(0,F\_SETFL,flags|O\_NONBLOCK);

n=read(0,buf,512);

usleep(1);

fcntl(0,F\_SETFL,flags);

return(n);

}

int main(int argc,char\*\* argv){

int f = 40;

int x1, x2;

if(con::romax()%2 == 0)

mid=con::romax()/2;

else

mid = (con::romax()/2) + 1;

signal(SIGINT,interruptor);

signal(SIGWINCH, resize);

while(done < 1){

for(x1 = 1, x2 = con::romax(); x1 <= mid && x2 >= mid;x1 += 2,x2 -= 2){

for(int i = 0; i < 2; i++)

for(int j = 1; j <= con::comax(); j++){

cout << SGR(f) << CUP(x1+i,j);

cout << " " << flush;

cout << SGR(f) << CUP(x2-i,j);

cout << " " << flush;

usleep(1000);

}

if (f < 47) f++;

else f = 40;

}

if (kbin() > 0)

break;

}

cout << CUP(1,1) << SGR(0) << con::ED << flush;

}

**Приложение 2**

Результат работы программы

