*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ \_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_Компьютерные Системы и сети (ИУ6)\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по практической работе № 2**

**Название домашней работы: Создание программной системы с элементарным интерфейсом консольного режима на С++**

**Дисциплина: Объектно-ориентированное программирование**

Студент гр. ИУ6-22Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. П. Плютто**



(Подпись, дата) (И.О. Фамилия)



Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. А. Веселовская**

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

**Задание:**

Выполнить структурную декомпозицию, разработать структурную схему и алгоритмы программ. Реализовать на С++ в консольном режиме. Предусмотреть примитивный интерфейс типа меню, позволяющий выбирать нужную операцию.

”Угадай число”. Программа “задумывает” трехзначное число восьмеричной, десятичной или шестнадцатеричной систем счисления и рисует на экране три прочерка. Играющий за один ход вводит одну цифру. Если она входит в запись числа, то проставляется вместо прочерка. Необходимо угадать число за как можно меньшее количество попыток. Реализовать класс-шаблон, реализующий логику игры. Параметром шаблона должен быть класс, обеспечивающий использование чисел соответствующей системы счисления.

**Решение:**

Для интерфейса меню я решил использовать цветовую палитру терминала и управление при помощи кнопок клавиатуры. Цветовую палитру удалось реализовать при помощи вывода в терминал особых команд, вот их список:

Команды для установки цвета текста в терминале обычно используют коды ANSI Escape, которые начинаются со специальной последовательности символов "\033[".

Ниже представлены некоторые общие команды для терминала в Linux/Unix системах и соответствующие им цвета:

**Команда Описание Цвет**

\033[0m Сброс цвета Без цвета

\033[1;31m Жирный красный Ярко-красный

\033[1;32m Жирный зеленый Ярко-зеленый

\033[1;33m Жирный желтый Ярко-желтый

\033[1;34m Жирный синий Ярко-синий

\033[1;35m Жирный фиолетовый Ярко-фиолетовый

\033[1;36m Жирный голубой Ярко-голубой

\033[1;37m Жирный белый Белый

\033[0;31m Красный Темно-красный

\033[0;32m Зеленый Темно-зеленый

\033[0;33m Желтый Темно-желтый

\033[0;34m Синий Темно-синий

\033[0;35m Фиолетовый Темно-фиолетовый

\033[0;36m Голубой Темно-голубой

\033[0;37m Белый Серый

Так же использовал изменение цвета фона текста:

**Команда Описание Цвет фона**

\033[0m Сброс цвета Без цвета

\033[41m Красный Красный

\033[42m Зеленый Зеленый

\033[43m Желтый Желтый

\033[44m Синий Синий

\033[45m Фиолетовый Фиолетовый

\033[46m Голубой Голубой

\033[47m Белый Белый

\033[40m Черный Черный

*class* Color

{

*public:*

*void* setCyanBackground()

{

cout << "\033[46m";

}

*void* resetColor()

{

cout << "\033[0m";

}

*void* printRed(std::string str)

{

cout << "\033[1;31m" << str << "\033[0m";

}

*void* printGreen(std::string str)

{

cout << "\033[1;32m" << str << "\033[0m";

}

*void* printYellow(std::string str)

{

cout << "\033[1;33m" << str << "\033[0m";

}

*void* printBlue(std::string str)

{

cout << "\033[1;34m" << str << "\033[0m";

}

*void* printMagenta(std::string str)

{

cout << "\033[1;35m" << str << "\033[0m";

}

*void* printCyan(std::string str)

{

cout << "\033[1;36m" << str << "\033[0m";

}

*void* print(std::string str)

{

cout << str;

}

};

Для чтения любых символов с клавиатуры (в том числе и стрелок, esc и тд) я хотел использовать сторонние библиотеки, но все такие библиотеки написаны в основном для windows и для другой архитектуры, поэтому они не могла скомпилироваться для arm. Поэтому на основе встроенных библиотек я написал дополнительную программу для отслеживания любого нажатия с клавиатуры.

*int* getch()

{

*int* ch;

*struct* termios t;

tcgetattr(STDIN\_FILENO, &t);

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

tcsetattr(STDIN\_FILENO, TCSANOW, &t);

ch = getchar();

t.c\_lflag |= ICANON | ECHO;

tcsetattr(STDIN\_FILENO, TCSANOW, &t);

*return* ch;

}

Класс для генерации числа:

*class* Number

{

*private:*

*int* randomInt(*int* min, *int* max)

{

random\_device rd;

mt19937 gen(rd());

uniform\_int\_distribution<> dis(min, max);

*int* randomNum = dis(gen);

*return* randomNum;

}

*public:*

string value;

*int* base;

Number()

{

stringstream ss;

srand(time(NULL));

base = randomInt(0, 2);

*switch* (base)

{

*case* 0:

ss << oct << randomInt(64, 512);

value = ss.str();

ss.str("");

*break*;

*case* 1:

value = to\_string(randomInt(100, 1000));

*break*;

*case* 2:

ss << hex << randomInt(256, 4096);

value = ss.str();

ss.str("");

*break*;

*default*:

value = "0";

*break*;

}

}

*void* print()

{

cout << "Number is: " << value << "\n";

}

};

Вот основной класс который создает меню, игру, и пишет правила игры.

*class* Game : Color, Number

{

*private:*

*int* getch()

{

*int* ch;

*struct* termios t;

tcgetattr(STDIN\_FILENO, &t);

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

tcsetattr(STDIN\_FILENO, TCSANOW, &t);

ch = getchar();

t.c\_lflag |= ICANON | ECHO;

tcsetattr(STDIN\_FILENO, TCSANOW, &t);

*return* ch;

}

*public:*

*int* mainGame()

{

*char* ch;

Number n;

string us;

string nums{"0123456789abcdef"};

*bool* n1{}, n2{}, n3{};

*do*

{

system("clear");

printMagenta("Загаданное число: ");

*if* (n1)

printGreen(string{1, n.value[0]});

*else*

printRed("\_");

*if* (n2)

printGreen(string{1, n.value[1]});

*else*

printRed("\_");

*if* (n3)

printGreen(string{1, n.value[2]});

*else*

printRed("\_");

cout << "\n";

printMagenta("Основание числа: ");

cout << (n.base == 0 ? 8 : (n.base == 1 ? 10 : (n.base == 2 ? 16 : 0))) << "\n";

printYellow("Количество попыток: ");

cout << us.length() << "\n";

printBlue("Использованные числа: ");

*for* (*unsigned* *long* i = 0; i < us.length(); i++)

cout << us[i] << ' ';

cout << "\n";

*if* (not(n1 && n2 && n3))

printCyan("Введите число\n");

*else*

printGreen("Игра окончена\n");

printCyan("Нажмите ");

printRed("esc");

printCyan(" для выхода");

ch = getch();

*if* (ch == '\033')

{

*return* 1;

}

*if* (not(n1 && n2 && n3))

{

*for* (*unsigned* *long* i = 0; i < nums.length(); i++)

{

*if* (ch == nums[i])

{

us += ch;

*if* (n.value[0] == ch)

n1 = true;

*if* (n.value[1] == ch)

n2 = true;

*if* (n.value[2] == ch)

n3 = true;

}

}

}

} *while* (ch != 27);

*return* 0;

}

*int* mainRools()

{

system("clear");

printCyan("--------------------------------------------------------------------------------------\n");

printCyan("|");

printGreen(" Угадай число ");

printCyan("|\n|");

cout << " Программа “задумывает” трехзначное число восьмеричной, десятичной или шест- ";

printCyan("|\n|");

cout << " надцатеричной систем счисления и рисует на экране три прочерка. Играющий за ";

printCyan("|\n|");

cout << " один ход вводит одну цифру. Если она входит в запись числа, то ставится ";

printCyan("|\n|");

cout << " вместо прочерка. Необходимо угадать число за наименьшее количество попыток. ";

printCyan("|\n");

printCyan("--------------------------------------------------------------------------------------\n");

printCyan("Нажмите любую клавишу для выхода");

getch();

*return* 0;

}

Game()

{

mainMenu();

}

};

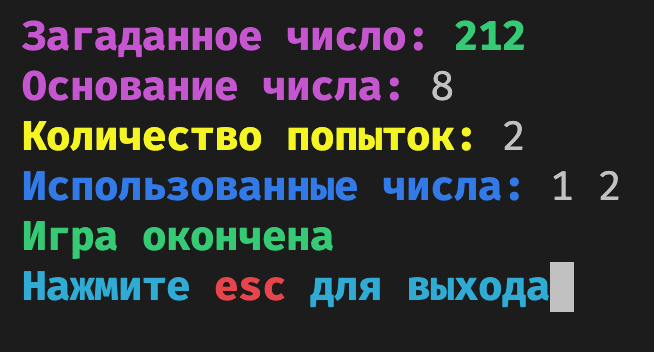
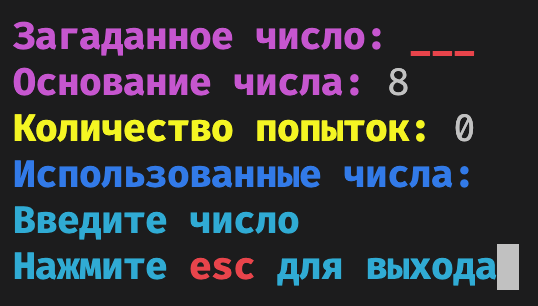
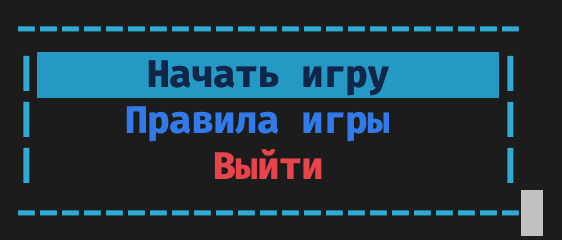
*int* main()

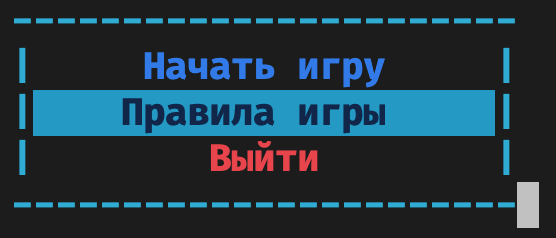
{

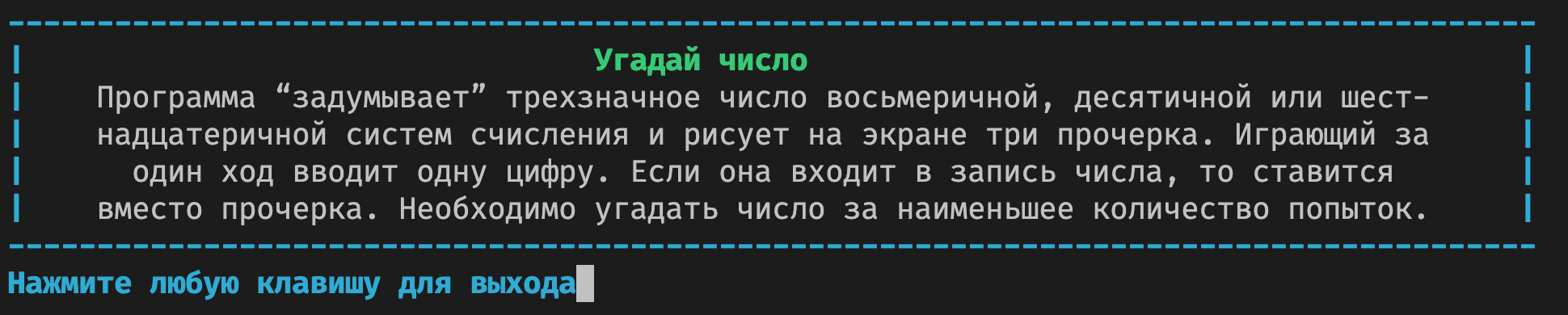
Game game;

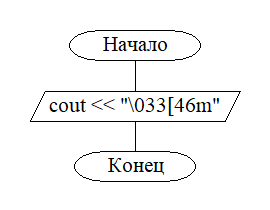
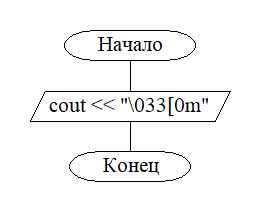
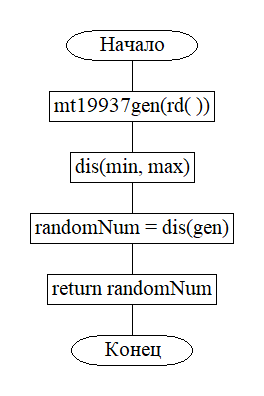
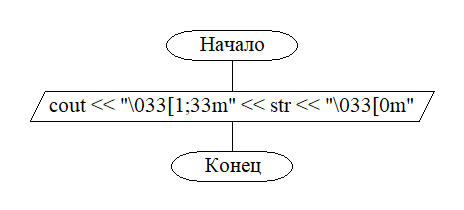
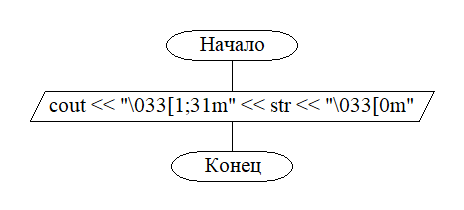
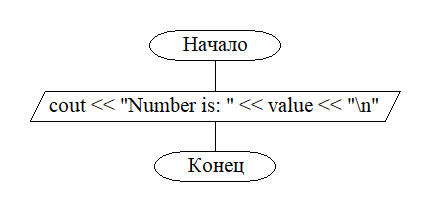
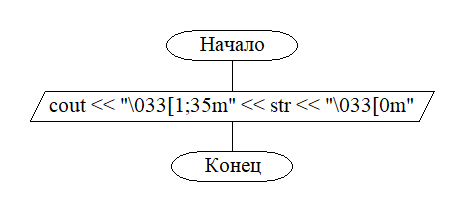
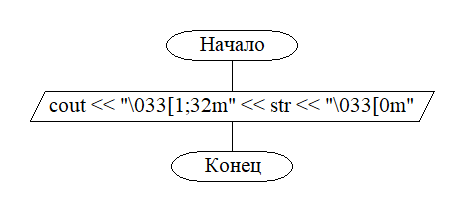
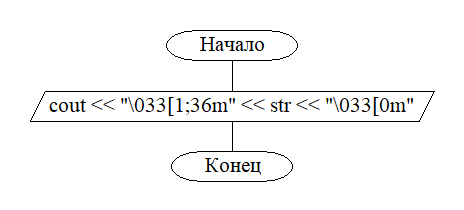
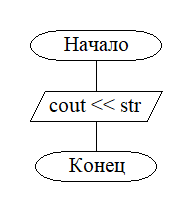
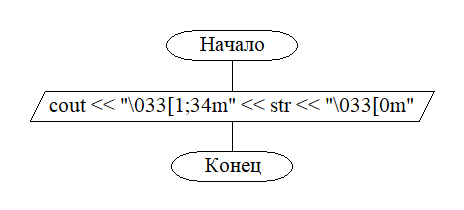
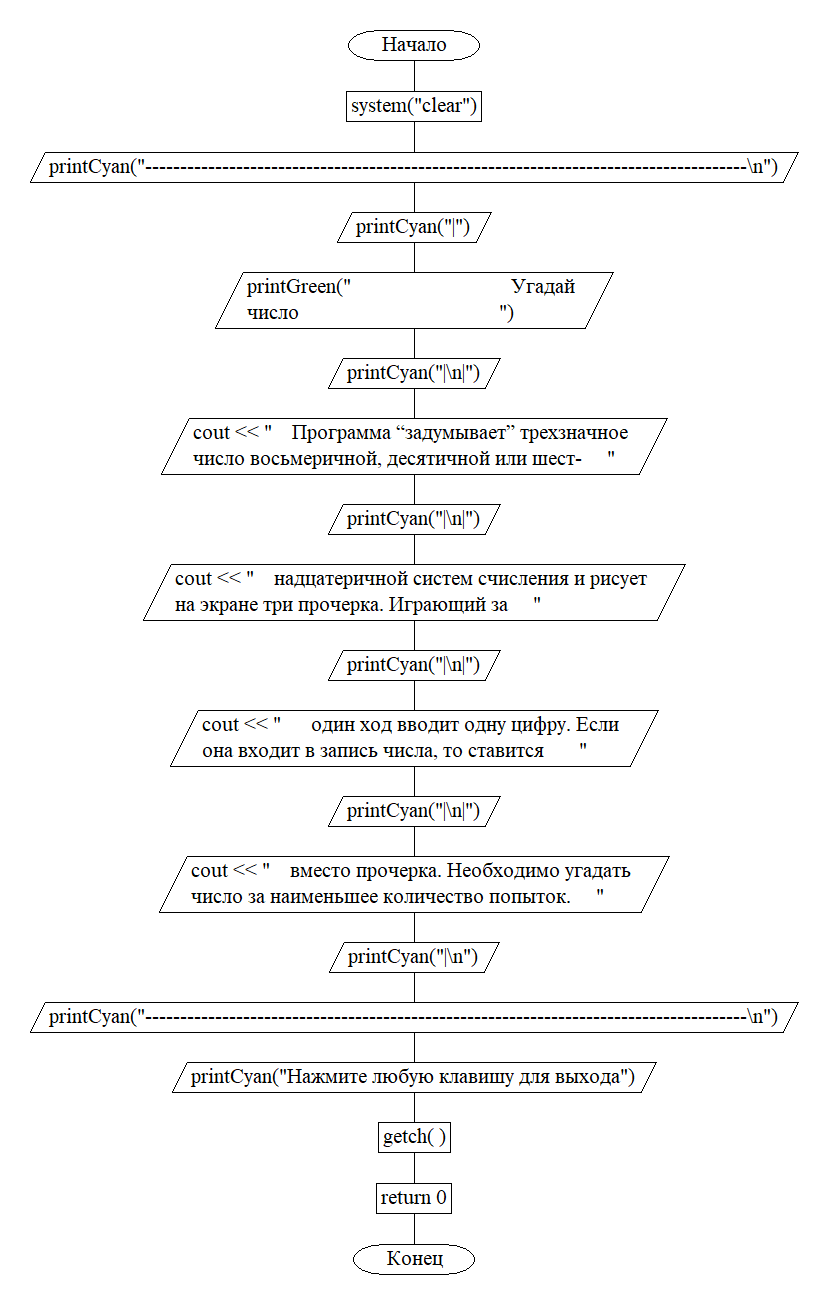
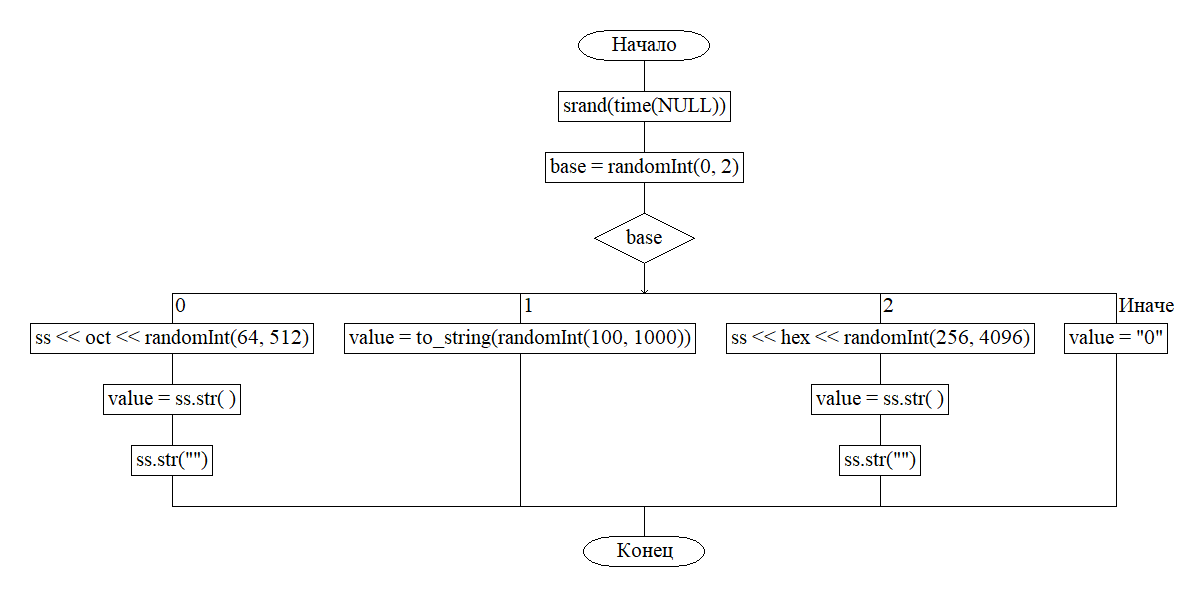
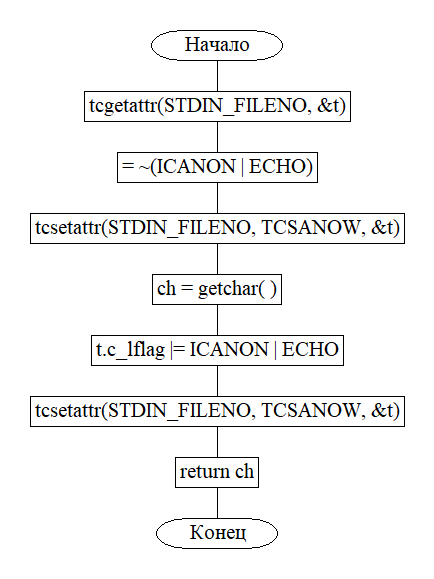
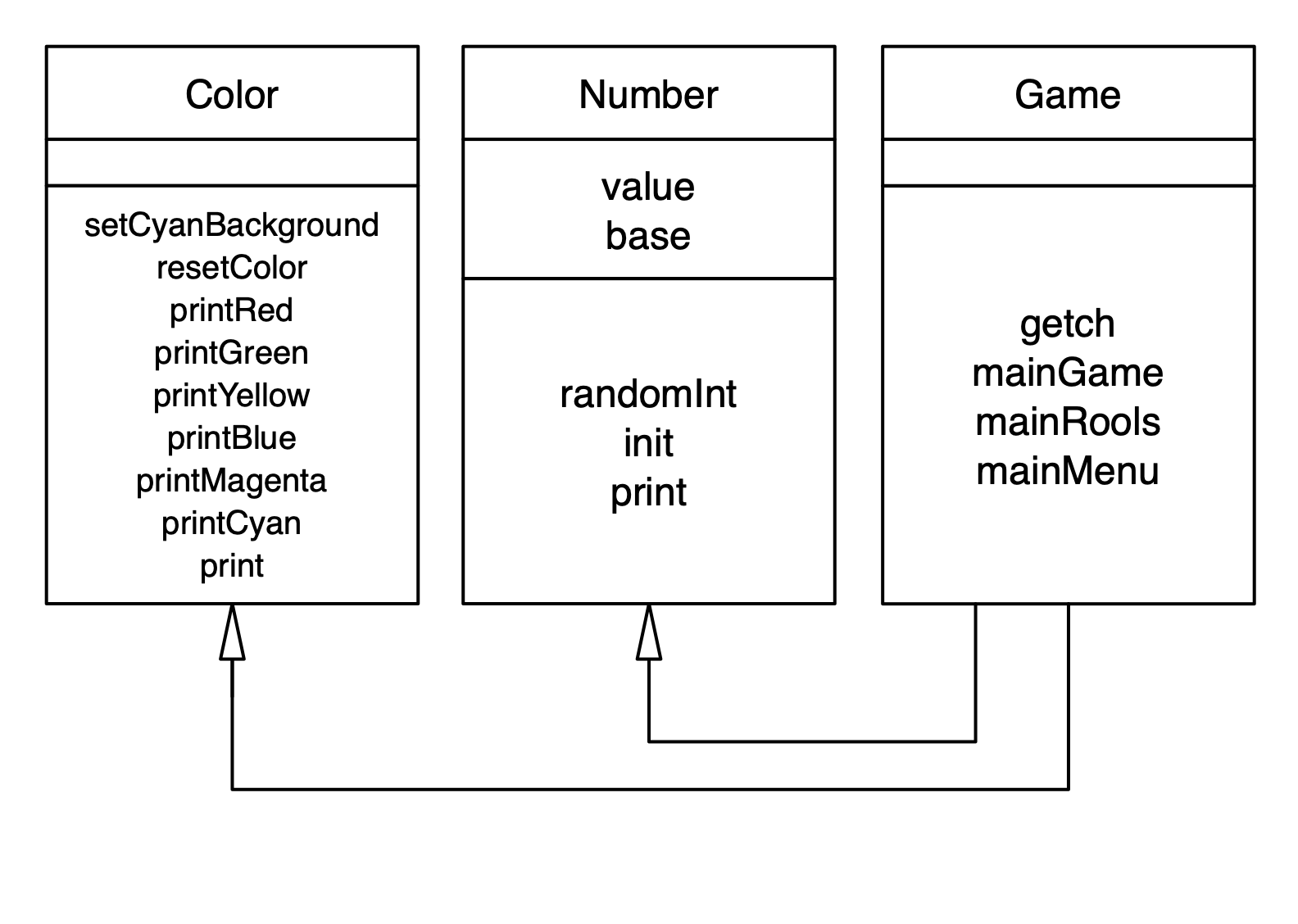
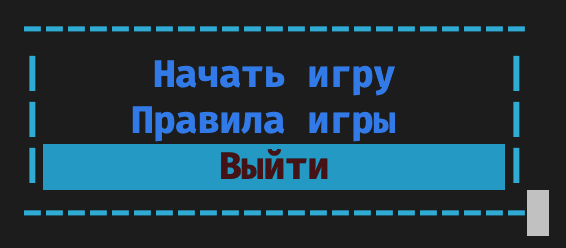
*return* 0;

}









Вывод: Я научился создавать стандартный интерфейс в терминале.