**1.Задача курсовой работы:**

Создать класс «Ипподром», клиент ставит на выбор на одну из нескольких лошадей, которые характеризуются: порода, кличка, год рождения, количество побед, а также случайным образом задаются скорости каждой лошади в скачках на каждый забег. Первые три параметра и количество лошадей задаются из внешнего файла. Во время скачек клиент наблюдает за бегами и по результатам узнает, какая лошадь какое место заняла; выводится сообщение выиграла ли текущий забег выбранная лошадь. По суммарным результатам пяти забегов сообщается - выиграл клиент или нет. Использовать паттерн «Наблюдатель» для реализации клиента. Результаты всех скачек и характеристики лошадей записываются в выходной файл. Если после пяти забегов будет сделаны новые ставки, то должен быть создан другой выходной файл и так далее.

**2.Функциональные требования (ФТ):**

2.1. Необходимо использовать паттерн «Наблюдатель»

2.2. Программа должна обеспечивать просмотр результатов скачек.

2.3. Заполнение первых трех характеристик лошадей выполняется через текстовый файл: File.txt.

2.4. Каждая лошадь обладает следующими характеристиками:

* порода
* кличка
* год рождения
* количество побед
* скорость

**3.Спецификации:**

3.1. При запуске программы на экране высвечивается окно, в котором пользователю предлагется ввести свое имя.

3.2 после выбора имени пользователю предлагается выбрать номер лошади, на которую он хочет сделать ставку (номер от 0 до 2).

3.3. После выбора имени и лошади в окне будет показан список забегов (от 0 до 4) в каждом из которых указана скорость каждой лошади и информация о выигрыше или проигрыше выбранной лошади.

3.4. Пользователь может нажать кнопку «у» для того чтобы выбрать другую лошадь

3.5 Пользователь может закончить ставки

3.6 Сохранение введенных данных в выходной файл. Формат файла описан в приложении.

**4. Программа методики испытаний.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Сценарий** | **Ожидаемый результат** | **№ФТ**  **№Спец-ии** | **Результат**  **№рисунка** |
| 1 | Запуск программы | Отображается сообщение о вводе имени | 3.1 | Рис. 1 |
| 2 | Ввели «lola» | Отображается сообщение о выборе лошади | 3.2 | Рис. 2 |
| 3 | Ввели «0» | Отображаются скачки, лошади и их скорости, результат скачек и предложение продолжения.  Все данные сохранились в выходной файл. | 3.3  3.4  3.5  3.6  2.2  2.1 | Рис. 3 |
| 4 | Ввели «у» | Программа снова запускается и отображается сообщение о вводе имени | 3.1 | Рис. 4 |
| 5 | Ввели «piter» | Отображается сообщение о выборе лошади | 3.2 | Рис. 5 |
| 6 | Ввели «2» | Отображаются скачки, лошади и их скорости, результат скачек и предложение продолжения.  Все данные сохранились в новый выходной файл. | 3.3  3.4  3.5  3.6  2.2  2.1 | Рис. 6 |
| 7 | Закрыли окно | Программа завершилась, все введенные данные сохранились в выходной файл | 3.4  3.5  3.6 | Рис. 7 |

**5.Текст программы**

#include <iostream>

#include <string>

#include <vector>

#include <fstream>

using namespace std;

class Ippo

{

protected:

class Observer \* views;

string por;

string name;

int year;

int win;

int speed;

bool winner;

public:

Ippo() {};

Ippo(string por, string name, int year)

{

this->por = por;

this->name = name;

this->year = year;

this->win = 0;

views = NULL;

}

void setspeed(int a)

{

this->speed = a;

}

int getspeed()

{

return this->speed;

}

void setwin()

{

this->win++;

}

int getwin()

{

return this->win;

}

void setwinner(bool flag)

{

this->winner = flag;

notify();

}

bool getwinner()

{

return this->winner;

}

void attach(Observer \*obs) {

views = (obs);

}

string myprint()

{

string res, res1;

res = "name: " + this->name + " poroda: " + this->por;

char\* tmp = new char(10);

\_itoa\_s(this->year, tmp, 4, 10

res1 = (string)tmp;

res += " year: " + res1;

\_itoa\_s(this->win, tmp, 4, 10);

res1 = (string)tmp;

res += " win: " + res1;

\_itoa\_s(this->speed, tmp, 4, 10);

res1 = (string)tmp;

res += " speed: " + res1;

return res;

}

void notify();

~Ippo() {};

};

class Observer {

Ippo \*model;

string o\_name;

public:

Observer(Ippo \*mod, string n) {

model = mod;

o\_name = n;

model->attach(this);

}

Observer() {

model = NULL;

}

void update()

{

bool win = getSubject()->getwinner();

cout << o\_name;

if (win)

{

cout << " win " << '\n';

}

else

cout << " not win " << '\n';

}

protected:

Ippo \*getSubject() {

return model;

}

};

void Ippo::notify() {

views->update();

}

int main()

{

vector<Ippo> myIppo;

ifstream in;

in.open("File.txt");

int kol, god;

string line, poroda, imya;

getline(in, line);

kol = atoi(line.c\_str());

int numst = 0;

bool contin=true;

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

{

getline(in, poroda);

getline(in, imya);

getline(in, line);

god = atoi(line.c\_str());

Ippo tmpHorse(poroda, imya, god);

myIppo.push\_back(tmpHorse);

}

in.close();

while (contin)

{

cout << "Vvedite name" << endl;

string tmp;

cin >> tmp;

cout << "Viberite loshad" << endl;

int l;

cin >> l;

Observer obs1(&myIppo[l], tmp);

for (int i = 0; i < 5; i++)

{

cout << i << endl;

for (int j = 0; j < kol; j++)

{

myIppo[j].setspeed(rand() % 50);

cout << j << " " << myIppo[j].getspeed() << endl;

}

int max = 0;

for (int j = 1; j < kol; j++)

{

if (myIppo[j].getspeed() > myIppo[max].getspeed())

{

max = j;

}

}

myIppo[max].setwin();

}

int max = 0;

for (int i = 1; i < kol; i++)

{

if (myIppo[i].getwin() > myIppo[max].getwin())

{

max = i;

}

}

if (l == max)

{

myIppo[l].setwinner(true);

}

else

myIppo[l].setwinner(false);

ofstream out;

char \*numstch = new char(2);

char Y;

\_itoa\_s(numst, numstch, 4, 10);

string res1 = (string)numstch;

string fname;

fname += "out" + res1 + ".txt";

out.open(fname);

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

{

out << myIppo[i].myprint() << endl;

}

out.close();

cout << "press y for continue" << endl;

cin >> Y;

if (Y == 'y')

{

contin = true;

}

else

contin = false;

numst++;

}

return 0;

}

**6.Скриншоты.**

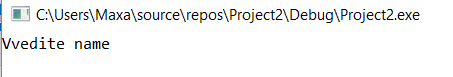
****

Рисунок 1. Запуск

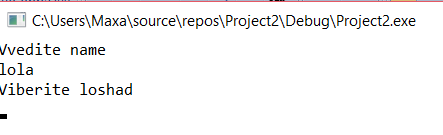


Рисунок 2. Ввод имени

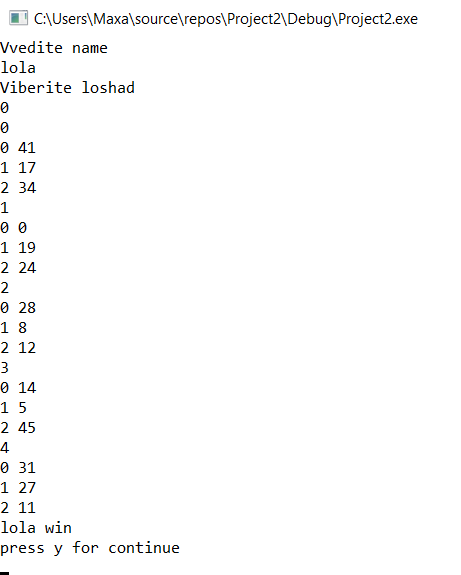


Рисунок 3. Просмотр результатов

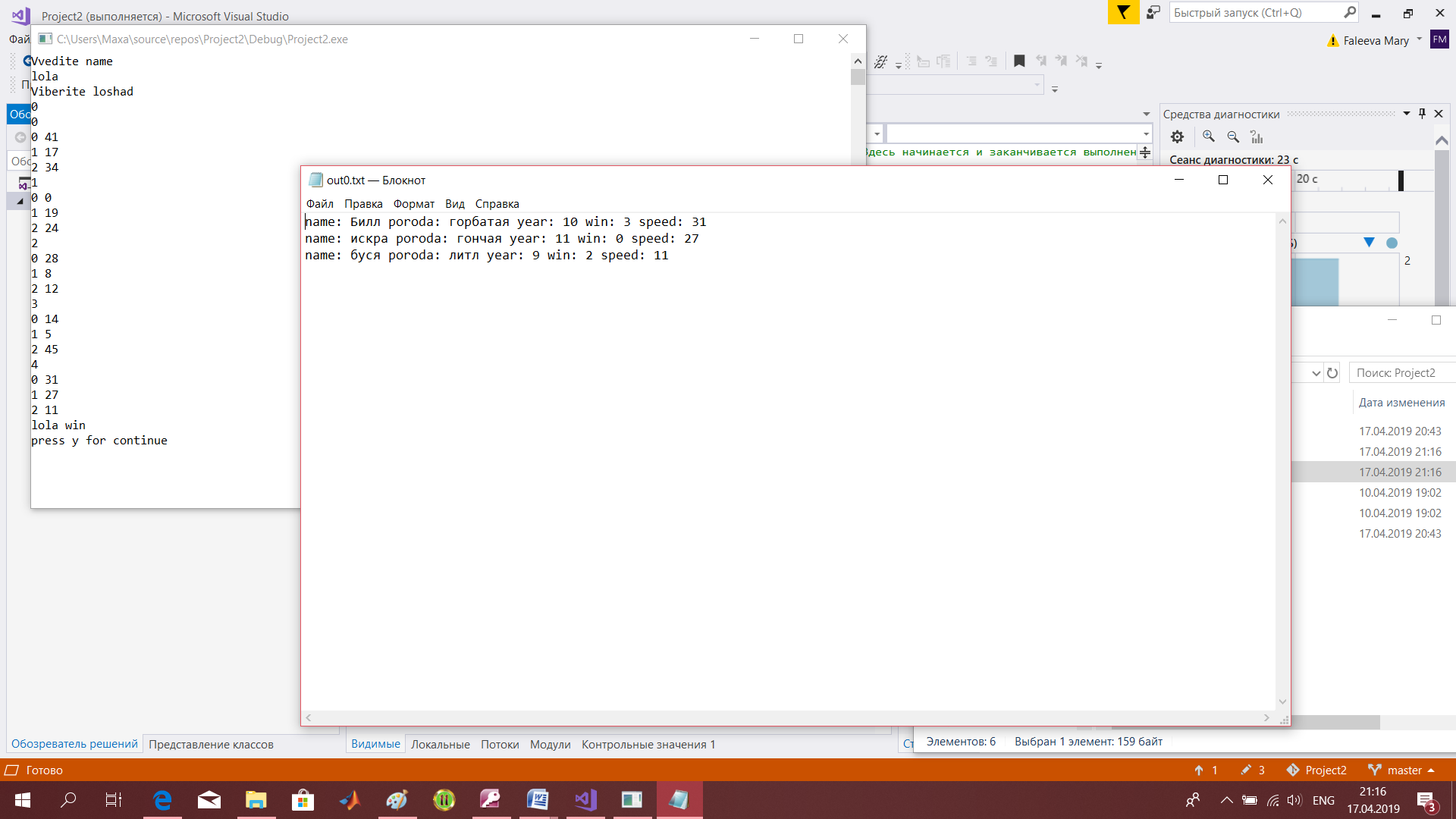
****

Рисунок 4. Сохранение данных в выходной файл

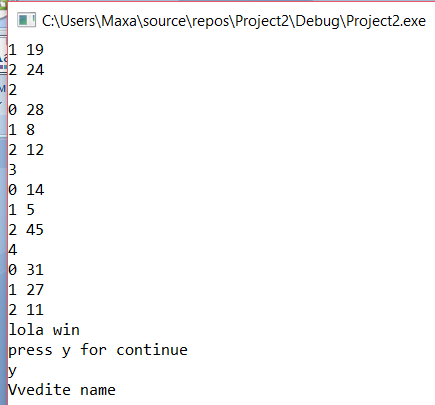


Рисунок 5. Продолжение ставок

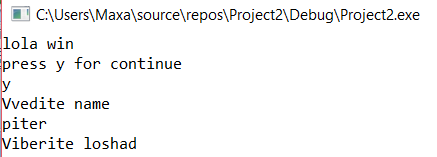


Рисунок 6. Ввод имени

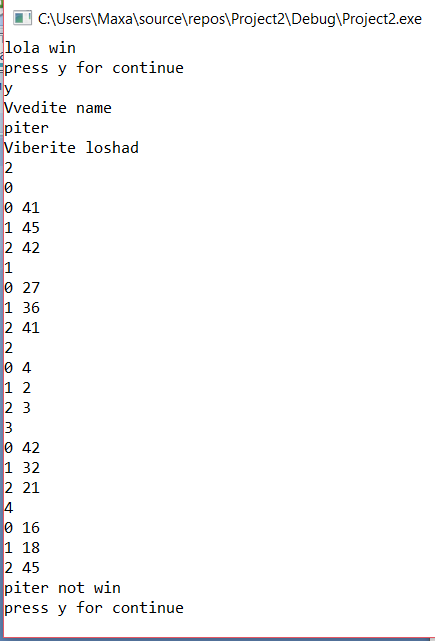


Рисунок 7. Просмотр результатов

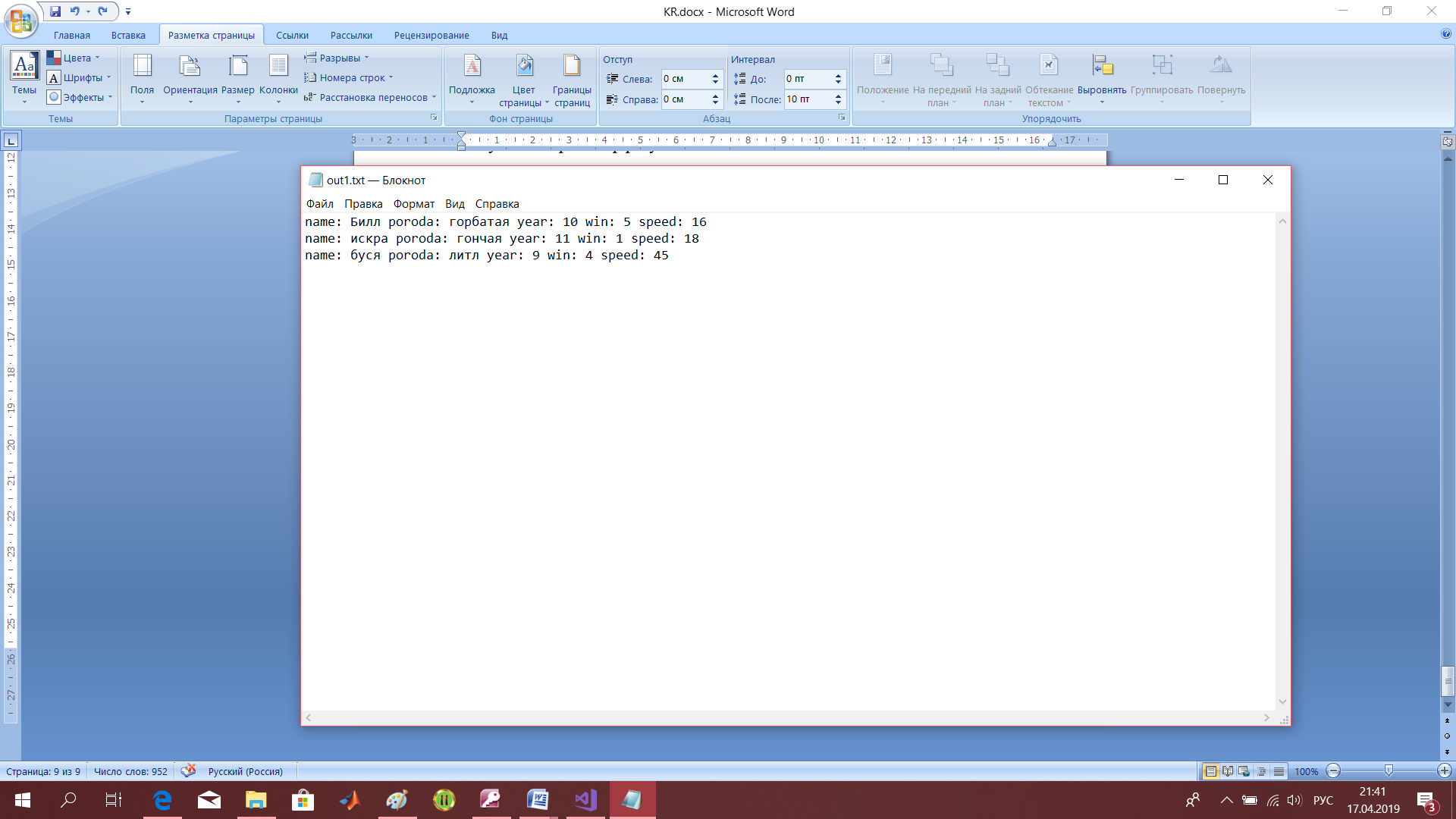


Рисунок 8. Сохранение данных в новый выходной файл

**7.Приложение.**

В данной работе используются файлы с расширением ".txt", указывающим на то, что это текстовый документ, который можно открыть в блокноте.

Файлы должны содержать обычный текст без форматирования и кодировку ANSI, состоящие из записей (строк), разделенных символами перевода строки, пробелами и запятыми.

Перед началом заполнения текстового файла, вводится количество лошадей. Данные об этом количестве выводится только на экран пользователю.

Характеристики лошадей заполняются из внешнего файла.

Порода лошади вводится в русской кодировке, также можно использовать английские символы. Максимально возможное количество символов: 255.

Кличка лошади вводится в русской кодировке, также можно использовать английские символы. Максимально возможное количество символов: 255.

Год рождения лошади заполняется с помощью цифр, максимальное возможное значение:9999 .

При каждой новой записи в программе данные перезаписываются.

Исходные файлы содержат данные о породе лошади, кличке и годе рождения и разделяются символом перевода строки.

Поля выходного файла содержат данные о породе лошади, кличке и годе рождения, а также о количестве побед и ее скорости.

Исходным файлом является: File.txt

Каждая характеристика лошади занимает одну строку.

Файлы с таким расширением являются исходной базой данных для лошадей.

Введенная информация с помощью программы сохраняется в файл "out.txt" таким образом, что сначала указывается кличка лошади, а после - все ее характеристики, скорость, и количество побед.

ГУАП

КАФЕДРА № 14

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | К.А. Курицын |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О КУРСОВОЙ РАБОТЕ |
|  |
| по курсу: Технологии программирования |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 1641 |  |  |  | М.С. Фалеева |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2019