



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

« МИРЭА Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Вычислительной техники

КУРСОВАЯ РАБОТА

по дисциплине

« Объектно-ориентированное программирование»

Наименование курсовой работы:

« К_1 »

С тудент группы

ИКБО-07-19

Ле Д..

Руководитель практики

Ассистент

Боронников А.С.

Работа представлена

«__»_____2020 г.

(подпись студента)

Оценка

(подпись руководителя)

Москва 2020

Постановка задачи

Дано поле 10 x 10 позиций. Нумерация позиций по горизонтали и по вертикали начинается с 1. Поле 10 x 10 формируется в файле field.txt и первоначально все позиции заполняются 8.

На вход подаются последовательно данные: координата позиции и символ для отметки. Координата позиции задается посредством пары целых чисел от 1 до 10. Первое число номер строки, второе номер столбца. Начало координат верхний левый угол поля. Символ для отметки принадлежит латинскому алфавиту.

Таких троек данных может быть множество. После получения очередной тройки **выдается сигнал** с текстом, содержащим координату и символ латинского алфавита. Если координата равна (0, 0), программа завершает работу. Первая строка входных данных не содержит (0, 0).

Использовать объекты:

1. Для установки позиции в файле согласно координате. Объект **выдает сигнал** об установке позиции или выводит в конце файла сообщение об ошибке.
2. Для записи символа в файле в установленную позицию. Объект пишет символ в установленной позиции или выводит в конце файла сообщение об ошибке.
3. Для вывода результата из файла на консоль. Все сообщения об ошибках пишутся построчно, с новой строки. Первое сообщение выводится с 11 строки (после поля 10 x 10).

- Написать программу, реализующую следующий алгоритм:
1. Создание файла и формирование исходного содержания.
 2. Ввод первой тройки данных.
 3. Начало цикла.
 - 3.1. Выдача сигнала о вводе данных.
 - 3.2. Ввод очередной тройки данных и выдача сигнала.
 4. Завершение цикла, если введена координата (0, 0).
 5. Вывод на консоль содержимого файла.

Описание входных данных

Построчно множество координат и символ:
«целое число»«целое число»«символ»

Описание выходных данных

Десять строк по десять символов в каждой, согласно сформированному в файле полю.

Если какая-либо координата не принадлежит интервалу [1, 10] то выводится сообщение:
Coordinate is wrong («значение», «значение»)

Если символ не принадлежит латинскому алфавиту:
Not a letter of the Latin alphabet: «символ»

Метод решения

Используя потоки Ввода/Вывода - cin/cout

Используя void buildFile для строить файл

Используя void createNewFile() для создать новый файл

Используя void scanPosition() для обрабатывать координаты

Используя int exes_app() для показать файл

Используя void printToScreen() для применять

Используя void setPosition(int x, int y, char k) для установки позиции в файле согласно координате.

Используя void insertChar(int x, int y) и void insertChar(char k, bool letter) для записи символа в файле в установленную позицию.

Используя void showFile() для вывода результата из файла на консоль.

Описание алгоритма

int main()

№ шага	Предикат	Действие	№ перехода
1		cl_application file_application; file_application.buildFile(); return file_application.exec_app();	Ø

void cl_base::setNameFile(string nameFile)

№ шага	Предикат	Действие	
1		this->nameFile = nameFile;	Ø

string cl_base::getNameFile()

1		return this->nameFile;	Ø
---	--	------------------------	---

void cl_base::setPosition(int x, int y, char k)

№ шага	Предикат	Действие	№ перехода
1	if (!(1 <= x && x <= 10)&&(1 <= y && y <= 10)))	addDataFile.open(nameFile, ios::app); insertChar(x, y);	Ø
	else		2
2	if (!(('a' <= k && k <= 'z') ('A' <= k && k <= 'Z')))	addDataFile.open(nameFile, ios::app); insertChar(k, false);	Ø
	else	addDataFile.open(nameFile, ios::in); int position = (x-1)*11 + y -1; addDataFile.seekp(position, ios::beg); insertChar(k, true);	Ø

void cl_base::insertChar(int x, int y)

№ шага	Предикат	Действие	№ перехода
1		addDataFile << endl << "Coordinate is wrong (" << x << ", " << y << ")"; addDataFile.close();	Ø

void cl_base::insertChar(char k, bool letter)

№ шага	Предикат	Действие	№ перехода
--------	----------	----------	------------

1	if (letter)	addDataFile.put(k); addDataFile.close();	Ø
	else	addDataFile << endl << "Not a letter of the Latin alphabet: " << k; addDataFile.close();	Ø

void cl_base::showFile()

№ шага	Предикат	Действие	№ перехода
1		readFile.open(nameFile, ios::in);	2
2	if (!readFile)	return;	Ø
	else		3
3		string line;	4
4	while(getline(readFile, line))	cout << line;	5
	0		6
5	if (!readFile.eof())	cout << endl;	4
	else		6
6		readFile.close();	Ø

cl_application::cl_application()

№ шага	Предикат	Действие	№ перехода
1		string name ="file.txt"; toDo.setNameFile(name);	Ø

void cl_application::buildFile()

№ шага	Предикат	Действие	№ перехода
1		createNewFile(); scanPosition();	Ø

void cl_application::createNewFile()

№ шага	Предикат	Действие	№ перехода
1		ofstream newFile(toDo.getNameFile(),	2

		ios::out);	
2	if (!newFile)	return;	Ø
	else		3
3	for (int indexR = 1; indexR <= 10; indexR++)		4
	indexR > 10;		6
4	if (indexR != 1)	newFile << endl;	5
	else		5
5	for (int indexC = 1; indexC <= 10; indexC++)	int number = 8; newFile << number;	5
	indexC > 10		3
6		newFile.close();	Ø

void cl_application::scanPosition()

№ шага	Предикат	Действие	№ перехода
1	while(true)	cin >> x >> y;	2
2	if (x == 0 && y == 0)	break;	Ø
	else	char k; cin >> k; todo.setPosition(x, y, k);	1

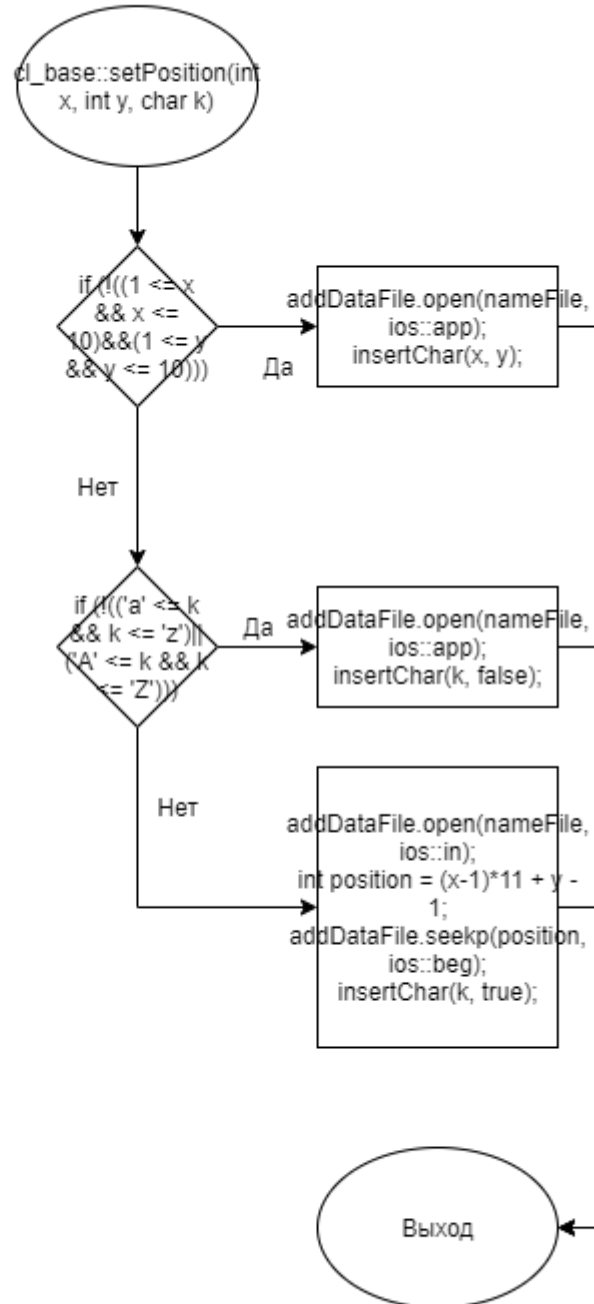
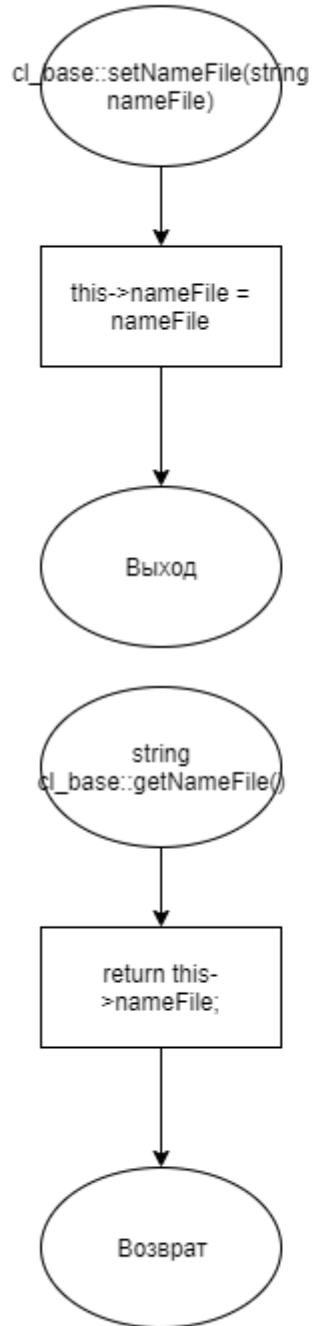
int cl_application::exec_app()

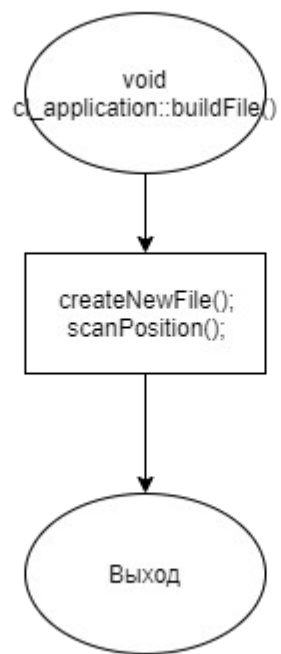
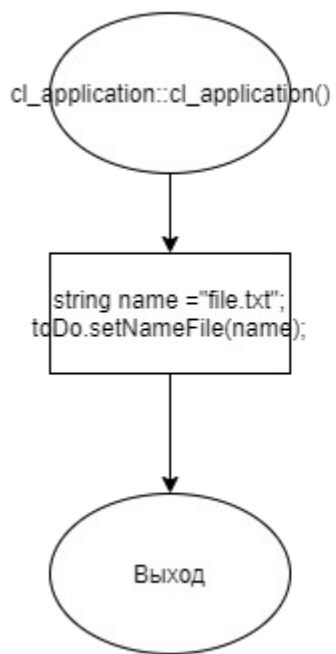
№ шага	Предикат	Действие	№ перехода
1		printToScreen(); return 0;	Ø

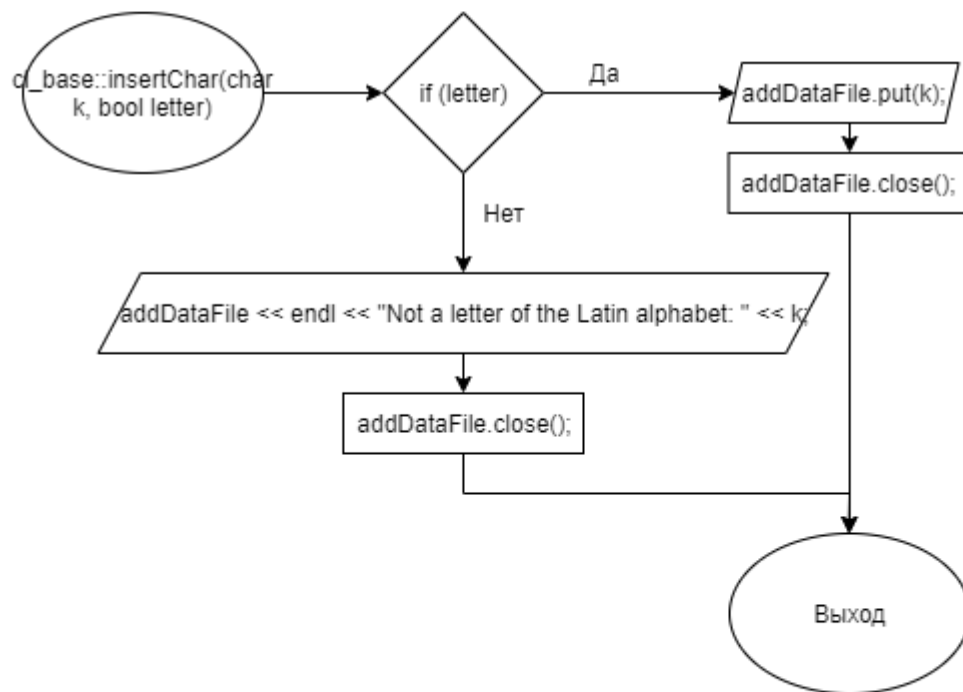
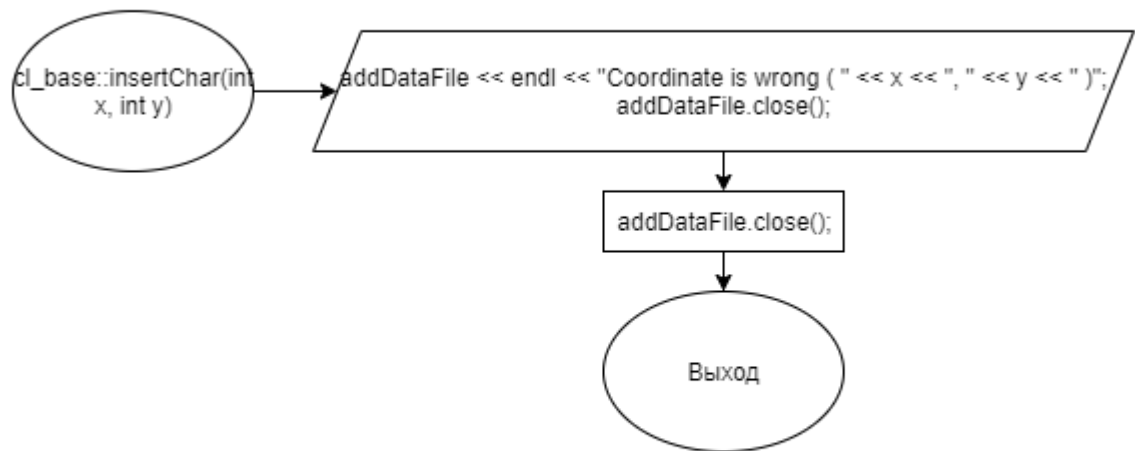
void cl_application::printToScreen()

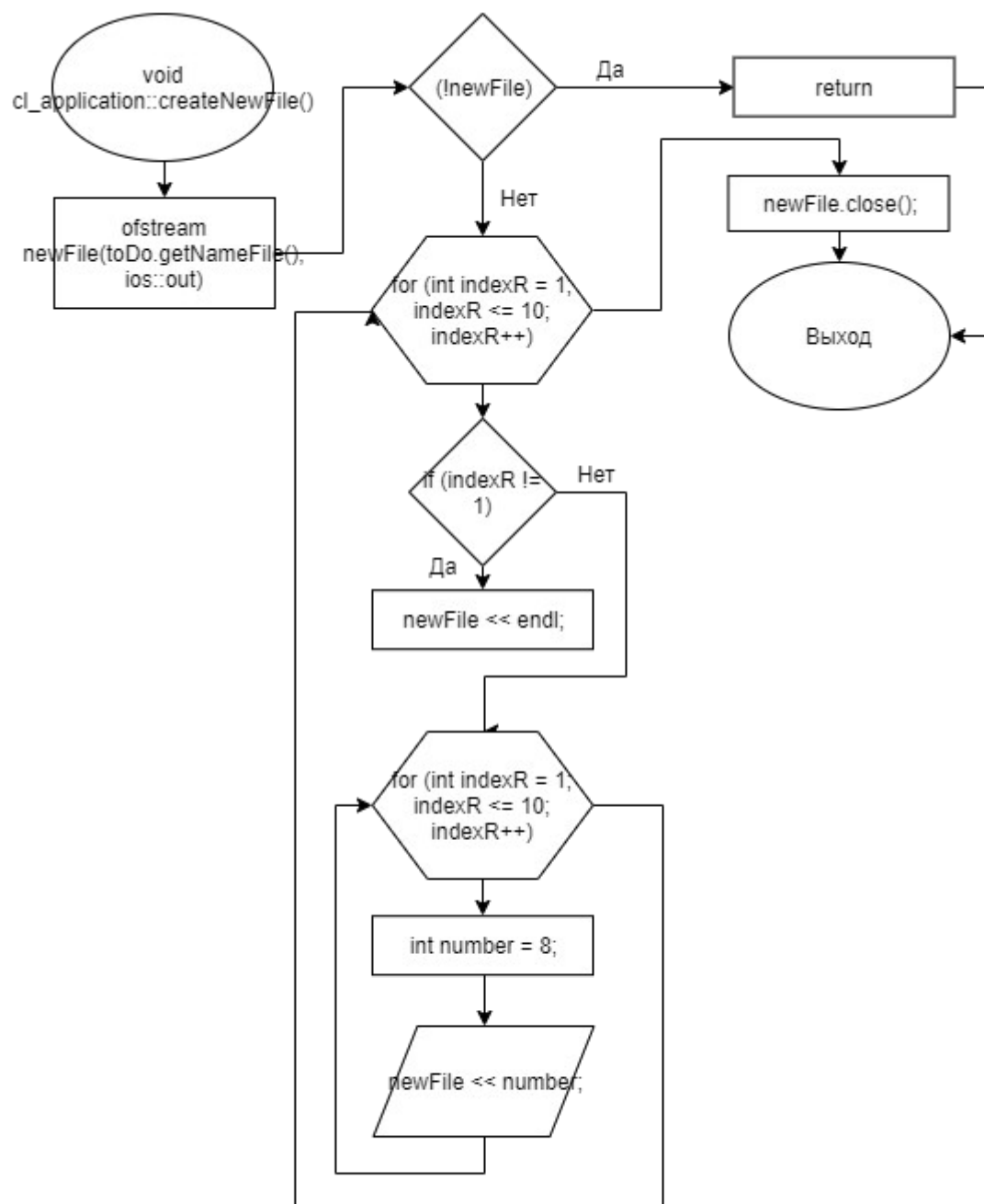
№ шага	Предикат	Действие	№ перехода
1		todo.showFile();	Ø

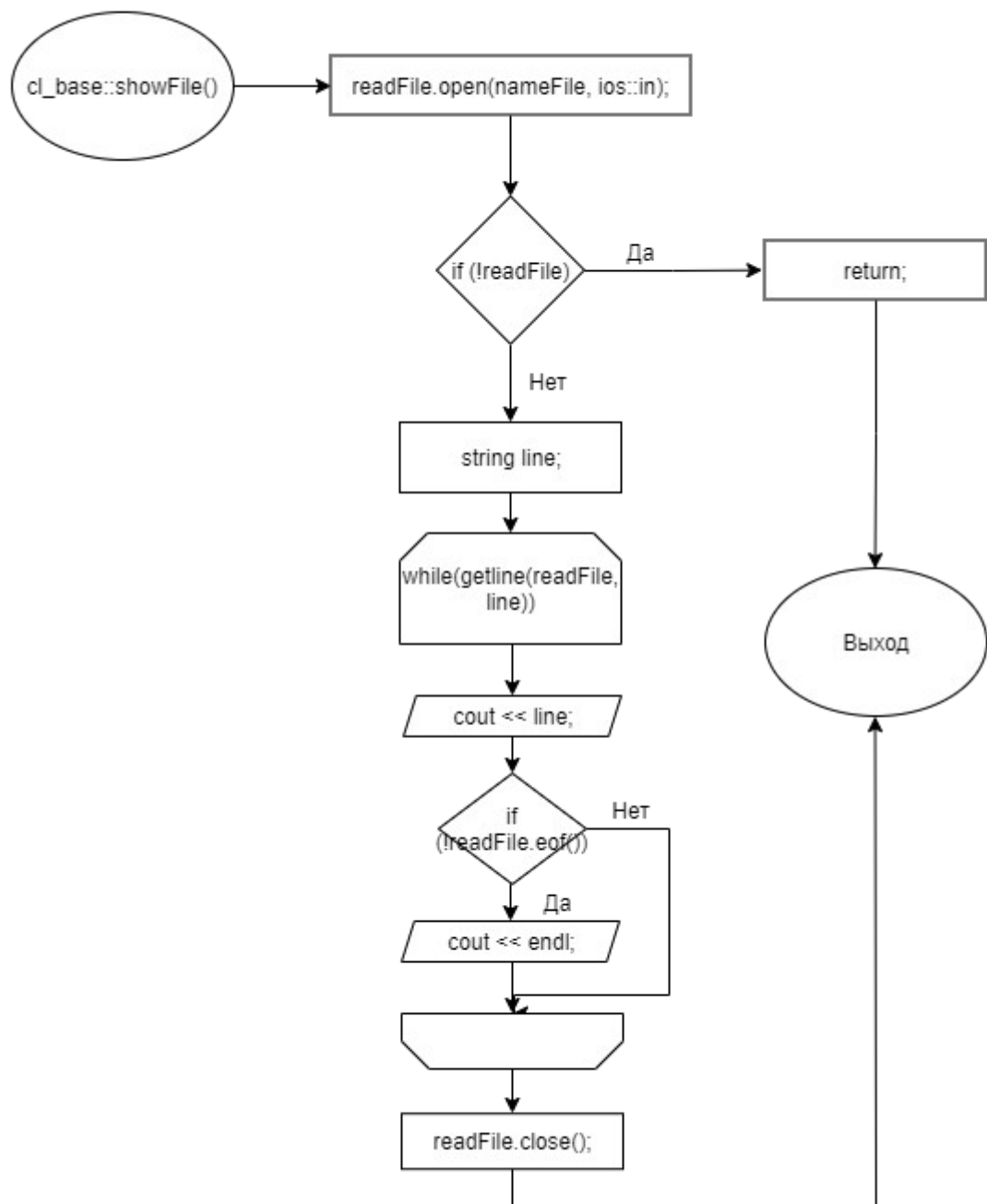
Блок-схема алгоритма

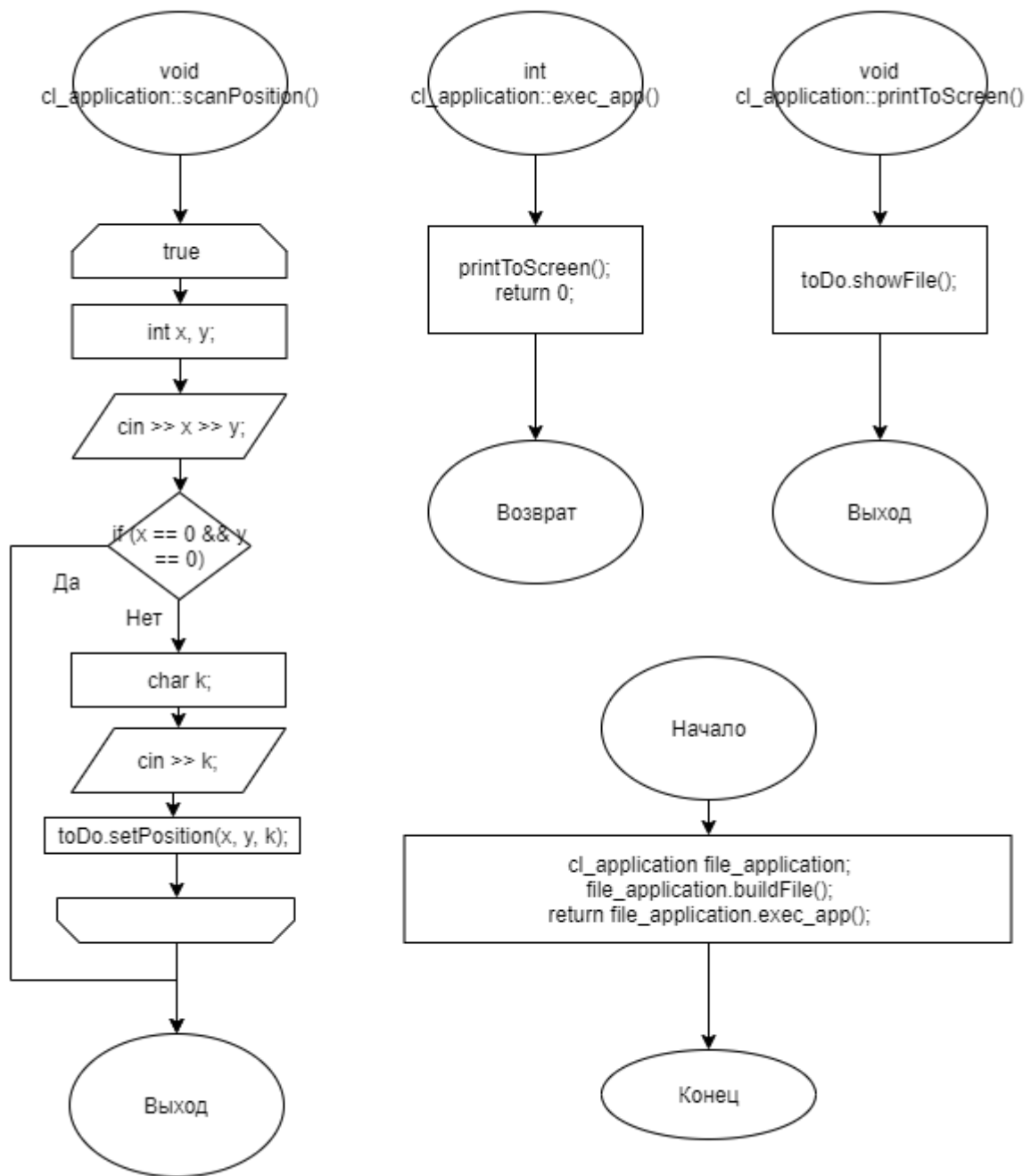












Код программы

Файл `cl_application.cpp`

```

#include "cl_application.h"
#include <iostream>
using namespace std;

cl_application::cl_application() {
    string name = "file.txt";
    toDo.setNameFile(name);
}

void cl_application::buildFile() {
    createNewFile();
    scanPosition();
}

void cl_application::createNewFile() {
    ofstream newFile(toDo.getNameFile(), ios::out);
    if (!newFile) {
        return;
    }
    for (int indexR = 1; indexR <= 10; indexR++){
        if (indexR != 1){
            newFile << endl;
        }
        for (int indexC = 1; indexC <= 10; indexC++){
            int number = 8;
            newFile << number;
        }
    }
    newFile.close();
}

void cl_application::scanPosition() {
    while(true){
        int x, y;
        cin >> x >> y;
        if (x == 0 && y == 0) {
            break;
        }
        else {
            char k;
            cin >> k;
            toDo.setPosition(x, y, k);
        }
    }
}

int cl_application::exec_app(){
    printToScreen();
    return 0;
}

void cl_application::printToScreen() {
    toDo.showFile();
}

```

Файл cl_application.h

```
#ifndef CL_APPLICATION_H
#define CL_APPLICATION_H

#include "cl_base.h"

class cl_application : public cl_base {
public:
    cl_application();
    void buildFile();
    void createNewFile();
    void scanPosition();
    int exec_app();
private:
    cl_base toDo;
    void printToScreen();
};

#endif //!CL_APPLICATION_H
```

Файл cl_base.cpp

```
#include "cl_base.h"
#include <iostream>
using namespace std;

void cl_base::setNameFile(string nameFile){
    this->nameFile = nameFile;
}
string cl_base::getNameFile() {
    return this->nameFile;
}

void cl_base::setPosition(int x, int y, char k){
    if (!((1 <= x && x <= 10)&&(1 <= y && y <= 10))){
        addDataFile.open(nameFile, ios::app);
        insertChar(x, y);
    }
    else {
        if (!(('a' <= k && k <= 'z')||('A' <= k && k <= 'Z'))){
            addDataFile.open(nameFile, ios::app);
            insertChar(k, false);
        }
        else {
            addDataFile.open(nameFile, ios::in);
            int position = (x-1)*11 + y - 1;
            addDataFile.seekp(position, ios::beg);
            insertChar(k, true);
        }
    }
}
```

```

        }
    }
}
void cl_base::insertChar(int x, int y){
    addDataFile << endl << "Coordinate is wrong ( " << x << ", " << y << "
)";
    addDataFile.close();
}
void cl_base::insertChar(char k, bool letter){
    if (letter) {
        addDataFile.put(k);
        addDataFile.close();
    }
    else {
        addDataFile << endl << "Not a letter of the Latin alphabet: "
<< k;
        addDataFile.close();
    }
}
void cl_base::showFile(){
    readFile.open(nameFile, ios::in);
    if (!readFile){
        return;
    }
    string line;
    while(getline(readFile, line)) {
        cout << line;
        if (!readFile.eof()) {
            cout << endl;
        }
    }
    readFile.close();
}
}

```

Файл cl_base.h

```

#ifndef CL_BASE_H
#define CL_BASE_H

#include <string>
#include <fstream>
using namespace std;
class cl_base {
private:
    string nameFile;
    ofstream addDataFile;
    ifstream readFile;
public:
    void setNameFile(string nameFile);
    string getNameFile();

```

```

        void setPosition(int x, int y, char k);
        void insertChar(int x, int y);
        void insertChar(char k, bool letter);
        void showFile();
};

#endif //!CL_BASE_H

```

Файл main.cpp

```

#include <iostream>
using namespace std;
#include "cl_application.h"

int main()
{
    cl_application file_application;
    file_application.buildFile();
    return file_application.exec_app();
}

```

Тестирование

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1 1 h 6 6 g 1 6 a 6 1 b 9 5 C 5 9 D -1 3 k -6 3 g 3 4) 1 10 A 10 10 B 0 0	h8888a888A 8888888888 8888888888 8888888888 888888888D8 b8888g8888 8888888888 8888888888 8888C88888 888888888B Coordinate is wrong (-1, 3) Coordinate is wrong (-6, 3) Not a letter of the Latin alphabet:)	h8888a888A 8888888888 8888888888 8888888888 888888888D8 b8888g8888 8888888888 8888888888 8888C88888 888888888B Coordinate is wrong (-1, 3) Coordinate is wrong (-6, 3) Not a letter of the Latin alphabet:)

