# **Лабораторная работа №1**

# Функциональные требования к программе

Функционал 1

1. Заходим в программу
2. Выводим уровень сложности
3. Ввод уровня сложности пользователем (неправильно)
4. Вывод сообщения об ошибке

Функционал 2

1. Заходим в программу
2. Выводим уровень сложности
3. Ввод уровня сложности пользователем (правильно)
4. Выводим поле
5. Пользователь вводит координату хода (неправильные координаты)
6. Вывод сообщения об ошибке и ввод координаты ещё раз

Функционал 3

1. Заходим в программу
2. Выводим уровень сложности
3. Ввод уровня сложности пользователем (правильно)
4. Выводим поле
5. Пользователь вводит координату хода (за пределами поля)
6. Вывод сообщения об ошибке

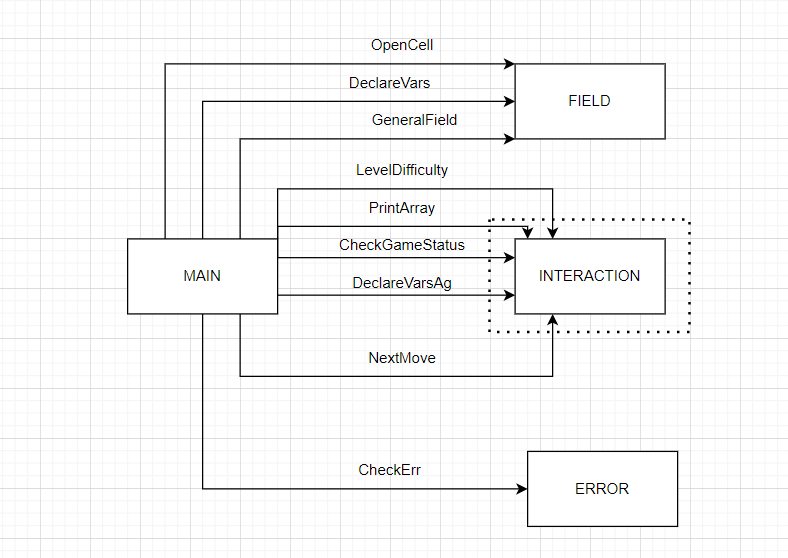
Функционал 4

1. Заходим в программу
2. Выводим уровень сложности
3. Ввод уровня сложности пользователем (правильно)
4. Выводим поле
5. Пользователь вводит координату хода (правильно)
6. П.6
7. Ввод координат мины
8. Вывод сообщения о конце игры (game over)
9. Конец игры. Вывод меню

Функционал 5

1. Заходим в программу
2. Выводим уровень сложности
3. Ввод уровня сложности пользователем (правильно)
4. Выводим поле
5. Пользователь вводит координату хода (правильно)
6. Поле изменяется (открытие клетки)
7. Ввод координаты (правильно)
8. П.6
9. Ввод координат до конца игры (до победы)
10. Вывод сообщения о конце игры (victory)
11. Конец игры. Вывод меню

# Структурная схема программы



# Прототипы реализации функций с комментариями

/\*\*

\*@brief DeclareVarsAg функция инициализации переменных

\*@param fieldH размер поля

\*@return 0 при успешном выполнении

\*/

int DeclareVarsAg (int fieldH);

/\*\*

\*@brief PrintField функция печати поля

\*@param array[][fieldHeight1] поле

\*@return -2 если пользователь вводит некорректное значение, 0 при успешном выполнении

\*/

int PrintField(int [][fieldHeight]);

/\*\*

\*@brief LevelDifficulty функция выбора уровня сложности

\*@return 0 если лёгкий, 1 если сложный, -1 если ошибка

\*/

int LevelDifficulty();

/\*\*

\*@brief NextMove функция перехода к следующему шагу

\*@param int[arr] массив координат

\*@return 0 при успешном выполнении

\*/

int NextMove(int arr[]);

# Команды компиляции исходных текстов и сборки библиотек

TARGET = minesweeper.exe

CC = gcc

${TARGET} : main.o field.o interaction.o checkErr.o

${CC} -o ./bin/${TARGET} ./obj/main.o ./obj/field.o ./obj/interaction.o ./obj/checkErr.o

main.o : main.c

${CC} -c main.c -o ./obj/main.o

field.o : ./field/field.c

${CC} -c ./field/field.c -o ./obj/field.o

interaction.o : ./interaction/interaction.c

${CC} -c ./interaction/interaction.c -o ./obj/interaction.o

checkErr.o : ./error/checkErr.c

${CC} -c ./error/checkErr.c -o ./obj/checkErr.o

clean:

del ./obj/\*.o ${TARGET}

# Выводы

# Содержание исходных кодов библиотек в приложении

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <time.h>

#include <stdbool.h>

#include <stdio.h>

#include <string.h>