

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ім. Ігоря Сікорського»
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

КУРСОВА РОБОТА

з дисципліни

Програмування та алгоритмічні мови

на тему: Японський кросворд

Студента 1 курсу групи КА-12

Спеціальність 123 Системний аналіз

Деменкова Олександра Віталійовича

Керівник ст. викладач Назарчук І.В.

Національна оцінка _____

Кількість балів: _____ Оцінка: ECTS ____

Члени комісії

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

Київ - 2022 рік

НТУУ „КПІ ім.І.Сікорського” ІІСА	
(назва вищого закладу освіти)	

Кафедра	<i>математичних методів системного аналізу</i>
---------	--

Дисципліна	<i>Програмування та алгоритмічні мови</i>
------------	---

Галузь знань	<i>12 Інформаційні технології</i>
--------------	-----------------------------------

Курс	<i>перший</i>	Група	<i>КА—12</i>	Семестр	<i>другий</i>
------	---------------	-------	--------------	---------	---------------

ЗАВДАННЯ на курсовий проект(роботу) студента

Деменков Олександр Віталійович
(прізвище, ім'я, по батькові)

1. Тема проекту(роботи)	Японський кросворд

2. Строк здачі студентом закінченого проекту(роботи)	20.05.2022 р.
--	----------------------

3. Вихідні дані до проекту(роботи)	Створити гру «Японський кросворд».
Роботу виконати в алфавітно-цифровому режимі консольного вікна з використанням керуючих клавіш.	

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)	
<i>1. Постановка задачі.</i>	
<i>2. Метод розв'язку задачі</i>	
<i>3. Загальна блок-схема алгоритму та опис алгоритму</i>	
<i>4. Опис програмного продукту.</i>	
<i>5. Результати роботи.</i>	
<i>6. Висновки.</i>	
<i>7. Список літератури.</i>	
<i>Додаток А. Текст програми.</i>	

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)	
<i>1. Загальна блок-схема алгоритму.</i>	
<i>2. Ілюстрації роботи програми.</i>	

6. Дата видачі завдання	
-------------------------	--

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент		
	(підпис)	

Керівник		Назарчук Ірина Василівна
	(підпис)	(прізвище, ім'я, по батькові)

(дата)	

Анотація

Представлена курсова робота є програмною реалізацією гри «Японський кросворд». Вона виконана за допомогою псевдо графічних елементів. При написанні курсової роботи використовувались мова програмування C++, псевдографіка здійснювалася на основі графічної бібліотеки <windows.h>.

Annotation

The presented course work is a software implementation of the package of solutions of systems of linear equations. C ++ programming language was used in writing the course work, pseudo-graphics was based on the graphics library > windows.h>.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 Постановка задачі.....	8
1.1 Огляд існуючих підходів до розв’язання поставленої задачі	8
1.2 Уточнена постановка задачі на розробку програмного забезпечення...	9
РОЗДІЛ 2 Розробка програмного продукту.....	10
2.1 Метод розв’язку задачі.....	10
2.2 Алгоритм розв’язку задачі.....	11
РОЗДІЛ 3 Опис розробленого програмного продукту.....	21
3.1 Опис головних структур і змінних програми.....	21
3.2 Опис головних функцій програми.....	22
3.3 Опис інтерфейсу.....	23
3.4 Результати роботи програмного продукту.....	25
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	29
Додаток А Текст програми.....	30

ВСТУП

Кожен з нас у дитинстві грав у ігри, хтось грав на комп'ютері хтось на телефоні, хтось у те, що придумав з друзями. І однією з цих ігор була та все ще залишається гра, що являється не тільки для розваг, а й для розвитку логічного мислення. Тому оскільки ця гра все ще є актуальною, виникла ідея розробити унікальний аналог цієї гри. Цей проект було розроблено на основі вже існуючих варіантів.

Метою даної курсової роботи являється розробки програми-аналога гри «Японський кросворд» з можливістю вибору декількох варіантів рівнів.

Реалізація поставленої мети передбачала розв'язання таких задач:

1. На основі різних джерел визначити основні риси майбутньої розробки.
2. Розробити простий та зручний для користувача інтерфейс.
3. Скласти структурний алгоритм роботи майбутньої програми та реалізувати його в середовищі C++.

В даній курсовій роботі було визначено етапи її створення, розроблено унікальний принцип роботи програми.

Практичне значення цієї програми - допомога людині відволіктися від життєвих проблем. Регулярне розв'язування цієї головоломки поліпшує пам'ять, кмітливість, математичні здібності та увагу.

При написанні курсової роботи було використано середовище розробки DEV-C++, операційна система Windows, для знаходження інформації використовував вся браузер "Chrome". Для блок-схем я використовував "Draw io", а для підготовки та оформлення курсової роботи - Microsoft Word.

Розділ 1

ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд існуючих підходів до розв'язання поставленої задачі

Був обраний варіант з використання двомірного масиву та перевірки координат. Оскільки одна клітинка поля це не одиничний квадрат, потрібно ділити на довжину та висоту, щоб отримати вірні координати масиву. Отримавши елемент масиву, ми перевіряємо що ми отримали, сам масив складається з 0 та 1, де 1 це клітинка яку потрібно зафарбувати, а 0 «пуста», якщо ми отримуємо 1 то зафарбовуємо клітинку відповідно до положення курсора.

Переміщення курсора обмежене відносно поля гри. Є можливість ставити сірий хрестик у якійсь клітинці, якщо користувач впевнений що вона «пуста», у супротивному випадку буде виведено червоний хрестик.

Також існує масив для перевірки які клітинки вже заповнені, для того щоб не виникало накладання. Цей масив заповнюється 2 коли користувач робить помилку, а якщо правильно то 1. Завдяки цьому масиву виконується перевірка чи закінчив гравець гру чи ні.

Програма полягає у тому, що користувач обирає у меню один із рівнів та починає вирішувати японську головоломку.

Після виведення рамки користувачу потрібно почати вирішувати головоломку за допомогою цифр.

1.2 Уточнена постановка задачі на розробку програмного забезпечення

Постановка задачі полягає у створенні гри «Японський кросворд» та використанням псевдографіки.

Я розбив всю програму на певні функції з параметрами, які взаємодіють між собою. Це і функції, які обчислюють у якій клітинці знаходиться курсор, і функції псевдографіки, і функції малювання. Також

присутні функції переміщення курсору, залежно від того знаходимось ми в меню чи ні. Рух курсору виконано за допомогою клавіш.

Для того щоб замалювати квадрат, виконується переміщення курсора у лівий верхній кут, після чого за допомогою цикла в циклі виконується вивід символу білого кольору з зміщення курсора по координатам клітинки.

Присутня функція яка зчитує масив з файлу.

Розділ 2

РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1 Метод розв'язку задачі

Дана програма написана на мові C++ з використанням бібліотеки <windows.h> для створення псевдографіки, а також <conio.h>, <iostream>, <fstream>. У розділі 1 було вже сказано, основою цієї гри є набір функцій які відповідають за розв'язки, меню та графіку.

Для початку треба зробити меню, визначити положення курсора та самі пункти. Потім слід проробити функції для обчислення методів і вводу-виводу.

Також передбачене зручне меню, яке містить такі пункти:

- 1.Start - початок(при натисненні перенаправляє до наступного меню)
2. Rules - Правила до гри
3. Exit - вихід

Та пункти меню Рівнів:

1. Level1 (House)
2. Level3 (Fish)
3. Exit - вихід

2.2 Алгоритм розв'язку задачі

Наступна Блок-Схема ілюструє меню, яке зображено на Рисунку 2.1 та Рисунку 2.2

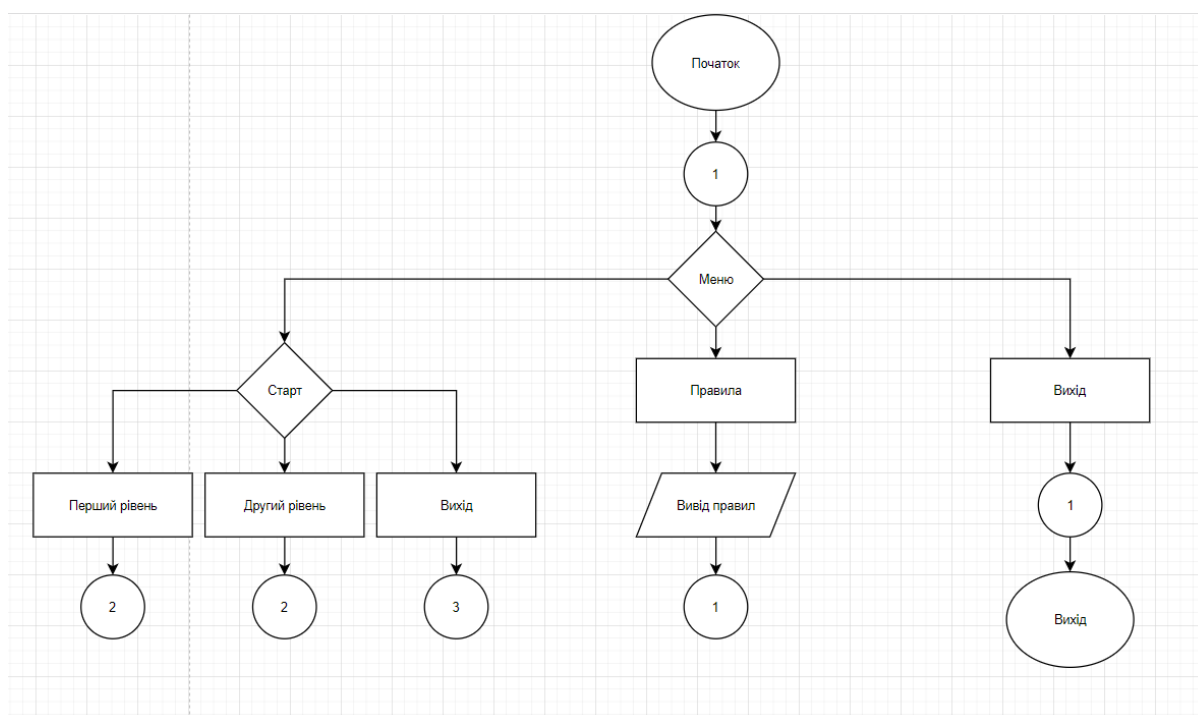


Рисунок 2.1 Структурна блок-схема меню

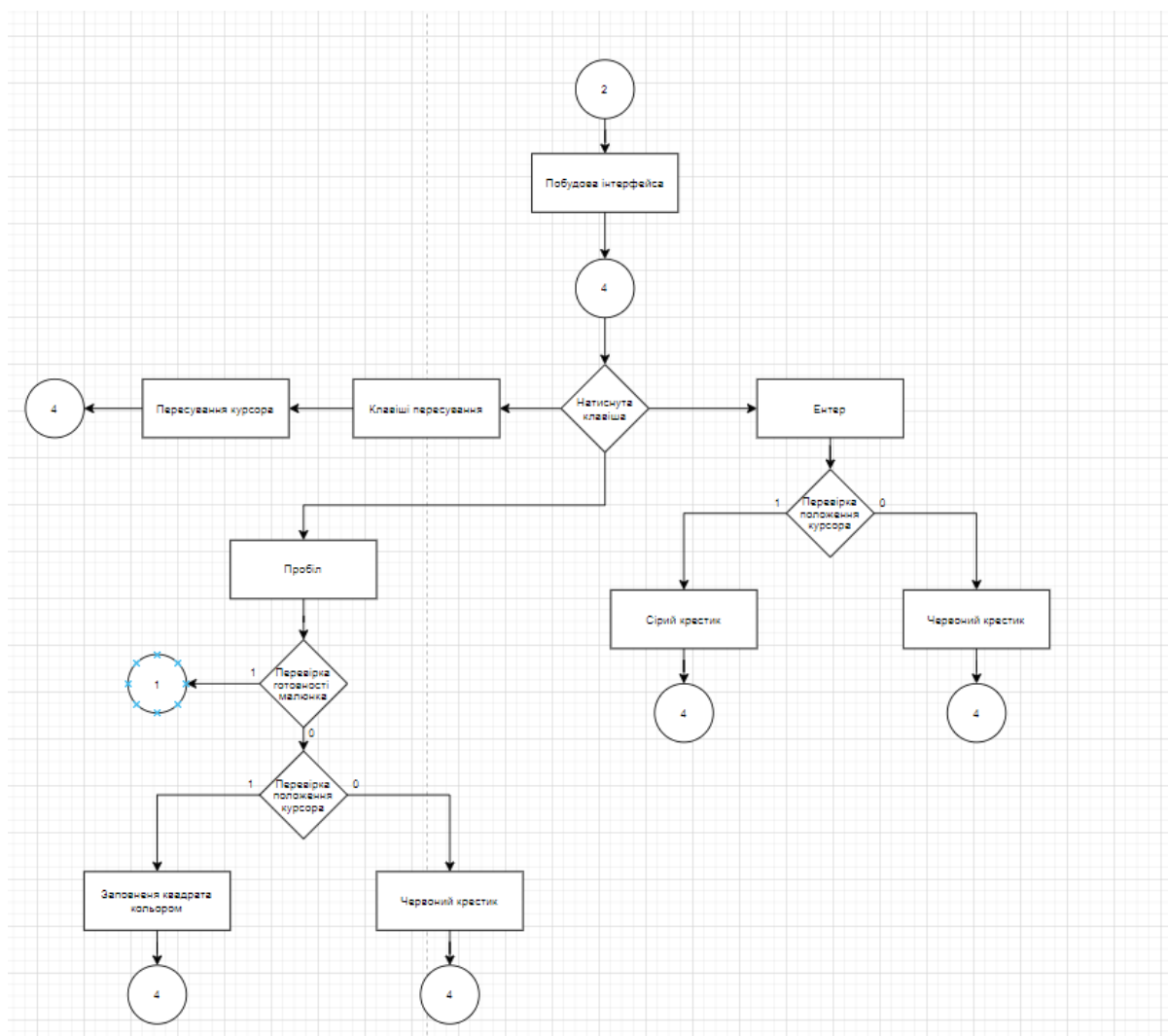


Рисунок 2.1 Структурна блок-схема меню

Розділ 3

ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ

3.1 Опис головних структур і змінних програми

Опис головних змінних програми, проілюстровано в Таблиці 3.1
найважливіші змінні

Таблиця 3.1 Опис головних змінних програми

№	Назва	Опис та призначення
1	bool playing	Перевірка чи запущена гра
2	bool escape	Перевірка для виходу
3	bool readingRules	Перевірка відкритих правил
4	int x, y	Задання початкових координат для курсора
5	int step	Крок переміщення курсора
6	char c	Натиснута клавіша
7	int k	Вибраний пункт меню
8	int n	Кількість пунктів меню
9	int pic[][]	Масив з малюнком
10	int filled[][]	Масив для перевірки які клітинки вже зафарбовані
11	int numbersR[][]	Масив цифр збоку
12	int numbersD[][]	Масив цифр знизу
13	COORD t, b;	Координати лівого верхнього та нижнього правого кутів

3.2 Опис головних функцій програми

Опис головних функцій програми продемонстровано в таблицях 3.2

Таблиця 3.2 Опис головних функцій програми

№	Назва	Опис та призначення
1	<code>void drawBond(HANDLE h, COORD t, COORD b, int numbersR[], int numbersD[])</code>	Малює поле гри
2	<code>void fill(HANDLE h, int x, int y)</code>	Заповнює один квадрат білим кольором на позиції курсора
3	<code>void cross(HANDLE h, int x, int y, int color)</code>	Малює крестик на позиції курсора
4	<code>void printMenu(HANDLE h, COORD c, int k, int n, int len)</code>	Виводить на екран меню
5	<code>void printLvls(HANDLE h, COORD c, int k, int n, int len)</code>	Виводить друге меню
6	<code>void printRules(HANDLE h)</code>	Виводить правила
7	<code>void cursorMove(HANDLE h, int x, int y, int step, int dirx, int diry)</code>	Переміщує курсор
8	<code>void read(int pic[], int numbersR[], int numbersD[], int number){</code>	Зчитує масив малюнка з файла

3.3 Опис інтерфейсу

На початку програми користувача зустрічає меню з пунктами Start, Rules, Exit. Це проілюстровано на Рисунку 3.1



Рисунок 3.1 Головне меню

На початку програми користувачу слід подивитися правила гри, якщо він їх ще не знає. На Рисунку 3.2 зображені правила

```
When solving Japanese crossword puzzles, a person considers each row / column separately, constantly moving on to the next columns and rows. In this case, the solution process in each row/column is reduced to:
1) Determining the cells that will definitely be painted over (for any possible arrangement of groups) - we paint over them.
2) Determination of cells in which the presence of filled cells is impossible - such cells are crossed out with a cross (sometimes a bold dot is used instead of a cross).
3) Identification of digits whose position has already been calculated - usually these digits are crossed out.
```

Рисунок 3.2 пункт “Rules”

Після натискання Esc або стрілочок користувач знову перейде до меню.

При натисненні пункта Exit, користувач виходить з програми. Це показано на рисунку 3.4

```
-----
Process exited after 574.8 seconds with return value 0
Press any key to continue . . . █
```

Рисунок 3.4 пункт “Exit”

При натисненні пункта “Start” користувача переводить до меню методів. Рисунок 3.5

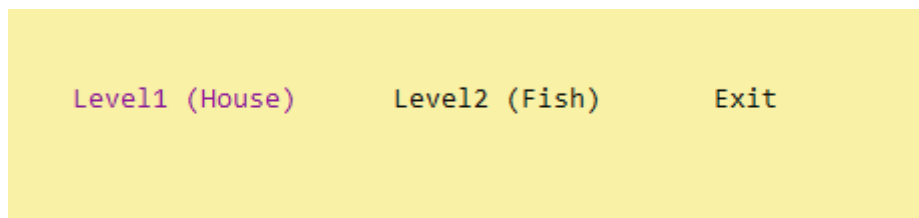


Рисунок 3.5 меню методів

3.4 Результати роботи програмного продукту

Коли користувач вибирає один із рівнів, завжди малюється поле для гри. Рисунок 3.6 та 3.7

															1
															2
															1
															2 2
															1 2
															9
															5 1 1 1
															7 1 1 1
															15
															1 1
															1 3 2 1
															1 3 2 2 1
															1 2 1
															1 2 1
															15
1 1	8	3 1	4 2 1	7 2 1	1 4 2 1	4 2 1	1 2 4	2 1 4	3 2 1	4 1 2 1	2 1 2 1	1 1 1 1	8	1 1 1	

Рисунок 3.6 Поле гри

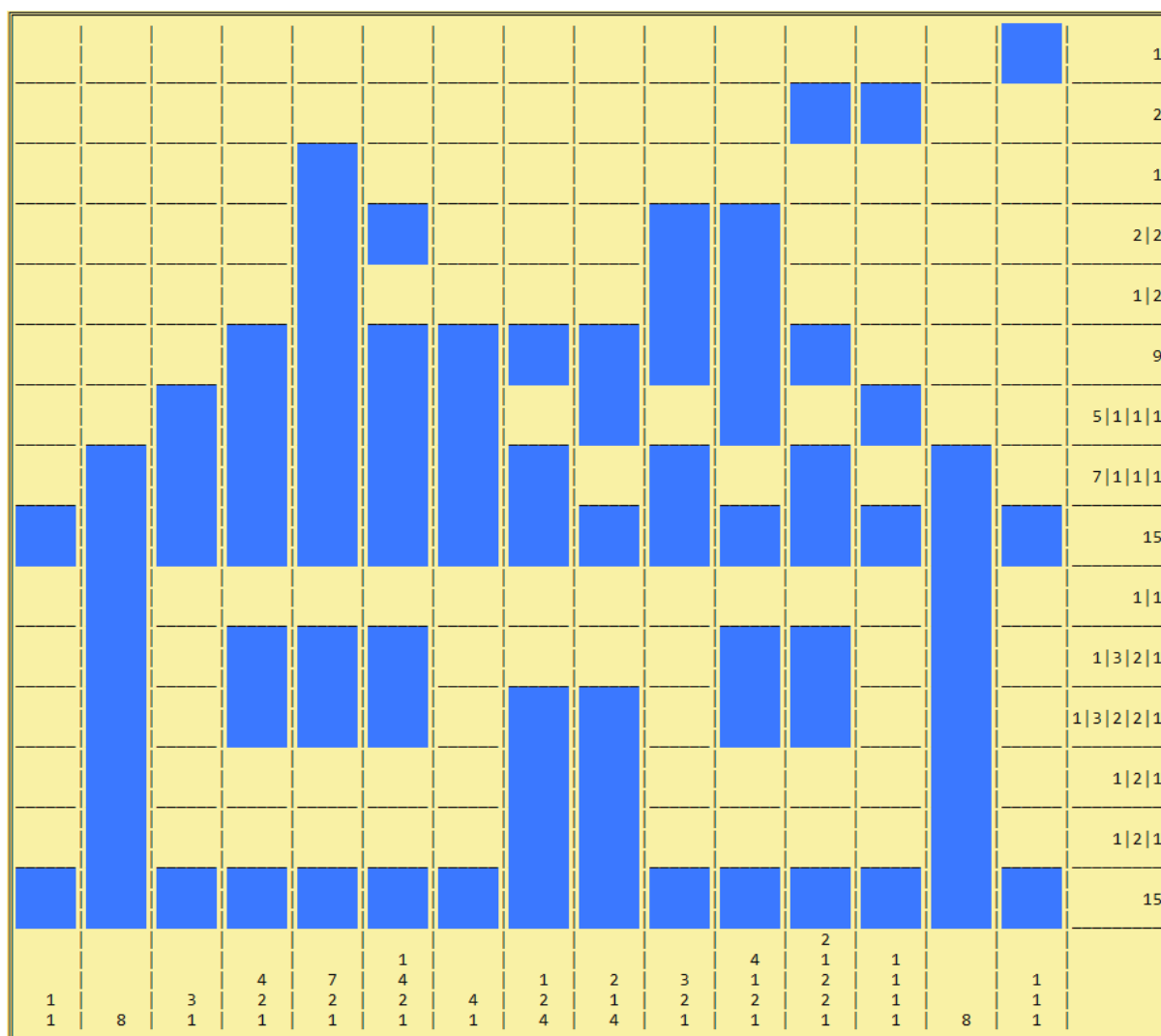


Рисунок 3.7 Вирішений кросворд

ВИСНОВКИ

Програма була написана за допомогою мови програмування C++ та графічної бібліотеки <windows.h>. У цій роботі були використані математичні підрахунки для визначення координат клітинки для її зафарбування. Також були використані методи порівняння елементів масивів для коректної перевірки проходження гри.

Перевага саме цієї програми в тому, що був розроблений простий та зручний для користувача інтерфейс. Також те, що можна доповнювати рівні в необмеженій кількості. Для вдосконалення розробленої програми можуть бути внесені правки щодо зміни інтерфейсу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Уклад.:І.В.Назарчук, Г.Г.Швачко – К.НТУУ «КПІ» 2017 -Методичні вказівки до виконання курсового проекту для студентів галузі знань 12 «Інформаційні технології» спеціальність 122 «Комп'ютерні науки та інформаційні технології», 124 «Системний аналіз».
2. Стенли Б. Л., Лажойе Ж.К., Барбара Э. М. Язык программирования C++: C++ Primer.
3. Шилдт Г.В. Книга C++: Повне керівництво.
4. Nonograms – Японські кросворди
URL: <https://www.nonograms.ru> (дата звернення: 29.05.2022).
5. Wikipedia – Японські кросворди
URL:https://uk.wikipedia.org/wiki/%D0%AF%D0%BF%D0%BE%D0%BD%D1%81%D1%8C%D0%BA%D0%B8%D0%B9_%D0%BA%D1%80%D0%BE%D1%81%D0%B2%D0%BE%D1%80%D0%B4 (дата звернення: 29.05.2022).

ДОДАТОК А

Текст програми

```
#include <Windows.h>
#include <conio.h>
#include <iostream>
#include <fstream>
using namespace std;

void drawBond(HANDLE h, COORD t, COORD b, int numbersR[15][5], int
numbersD[15][5]){
    system("cls"); SetConsoleTextAttribute(h, (WORD)((30 << 4) | 0));
    int lenx = b.X-t.X-2, leny = b.Y-t.Y-1;
    COORD f = t;
    unsigned char c;
    SetConsoleCursorPosition(h, f);
    c = 457; cout << c;
    c = 461;
    for(int i=0; i < lenx; i++){//top
        cout << c;
    }
    c=442; f=t; f.Y++;
    for(int i=0; i< leny-1;i++){//left
        SetConsoleCursorPosition(h, f);f.Y++;
        cout<<c;
    }
    SetConsoleCursorPosition(h, f);f.Y++;
```

```

c= 456; cout << c;
f.X = t.X+1; f.Y = b.Y; c= 461;
for(int i=0; i < lenx;i++){//bottom
    cout<<c;
}
c=444; cout<<c;
c=442;
f.X = b.X -1; f.Y = b.Y-2;
for(int i=leny; i >1;i--){//right
    SetConsoleCursorPosition(h, f);
    f.Y--;
    cout<<c;
}
f.X = b.X-1; f.Y = t.Y;
SetConsoleCursorPosition(h, f);
c=443; cout<<c;
c=95;
SetConsoleCursorPosition(h, { 1,2});
for(int i=1; i < 44; i+=3){//linesG
    SetConsoleCursorPosition(h, { 1,i+2});
    for(int j=0; j < lenx; j++){
        cout << c;
    }
}
c=124;
for(int i=7; i < 112; i+=7){//linesV
    SetConsoleCursorPosition(h, {i+2,1});
    for(int j=0; j < leny-1; j++){
        SetConsoleCursorPosition(h, {i,1+j});
    }
}

```

```

        cout << c;
    }
}
SetConsoleTextAttribute(h, (WORD)((30 << 4) | 0));
for(int i=0; i < 15; i++){
    for(int j=0; j < 5; j++){
        if(numbersR[i][j] >= 10)
            SetConsoleCursorPosition(h, {b.X - 10 + j*2 - 1, 2 +
i*3});
        else{
            SetConsoleCursorPosition(h, {b.X - 10 + j*2, 2 +
i*3});
        }
        if(numbersR[i][j] != 0){
            cout << numbersR[i][j];
            cout << "|";
        }
    }
}

for(int i=0; i < 15; i++){
    for(int j=0; j < 5; j++){
        SetConsoleCursorPosition(h, {t.X + 4 + i*7, b.Y - 6 + j});
        if(numbersD[i][j] != 0){
            cout << numbersD[i][j];
        }
    }
}
SetConsoleTextAttribute(h, (WORD)((30 << 4) | 0));

```

```

c=442;
f.X = b.X -1; f.Y = b.Y-2;
for(int i=leny; i >1;i--){//right
    SetConsoleCursorPosition(h, f);
    f.Y--;
    cout<<c;
}

```

```

    SetConsoleCursorPosition(h, {2, 2});
}

```

```

void fill(HANDLE h, int x, int y){
    SetConsoleTextAttribute(h, (WORD)((9 << 4) | 0));
    for(int i=0; i < 6; i++){
        for(int j=0; j < 3; j++){
            SetConsoleCursorPosition(h, {x+i-1, y+j-1});
            cout << " ";
        }
    }
}

```

```

void cross(HANDLE h, int x, int y, int color){//20 24
    SetConsoleTextAttribute(h, (WORD)((color << 4) | 0));
    for(int i=0; i < 3; i++){
        SetConsoleCursorPosition(h, {x+i*2-1, y+i-1});
        cout << " ";
        SetConsoleCursorPosition(h, {x+i*2, y+i-1});
        cout << " ";
    }
}

```



```

    for(int i = 3; i > 0; i--){
        SetConsoleCursorPosition(h, {x+i*2-3, y-i+2});
        cout << " ";
        SetConsoleCursorPosition(h, {x+i*2-2, y-i+2});
        cout << " ";
    }
}

char* menu[] = {"Start", "Rules", "Exit"};
char* lvls[] = {"Level1 (House)", "Level2 (Fish)", "Exit"};

void printMenu(HANDLE h, COORD c, int k, int n, int len){
    SetConsoleTextAttribute(h, (WORD)((30 << 4) | 5));
    system("cls");
    SetConsoleTextAttribute(h, (WORD)((30 << 4) | 0));
    COORD cc = c;
    for(int i=0; i<n; i++){
        cc.X = c.X + i*len;
        SetConsoleCursorPosition(h, cc);
        cout<< menu[i];
    }
    cc.X = c.X + k*len;
    SetConsoleTextAttribute(h, (WORD)((30 << 4) | 5));
    SetConsoleCursorPosition(h, cc);
    cout << menu[k];
}

void printLvls(HANDLE h, COORD c, int k, int n, int len){
    SetConsoleTextAttribute(h, (WORD)((30 << 4) | 5));
    system("cls");

```

```

SetConsoleTextAttribute(h, (WORD)((30 << 4) | 0));
COORD cc = c;
for(int i=0; i<n; i++){
    cc.X = c.X + i*len;
    SetConsoleCursorPosition(h, cc);
    cout << lvls[i];
}
cc.X = c.X + k*len;
SetConsoleTextAttribute(h, (WORD)((30 << 4) | 5));
SetConsoleCursorPosition(h, cc);
cout << lvls[k];
}

void printRules(HANDLE h){
    SetConsoleTextAttribute(h, (WORD)((30 << 4) | 5));
    system("cls");
    SetConsoleTextAttribute(h, (WORD)((30 << 4) | 0));
    cout << "When solving Japanese crossword puzzles, a person considers
each row / column separately, constantly moving on to the next columns and
rows. In this case, the solution process in each row/column is reduced to:";
    cout << endl << endl << "1) Determining the cells that will definitely be
painted over (for any possible arrangement of groups) - we paint over them." <<
endl;
    cout << "2) Determination of cells in which the presence of filled cells is
impossible - such cells are crossed out with a cross (sometimes a bold dot is
used instead of a cross)." << endl;
    cout << "3) Identification of digits whose position has already been
calculated - usually these digits are crossed out." << endl;
}

```

```

void cursorMove(HANDLE h, int x, int y, int step, int dirx, int diry){
    SetConsoleCursorPosition(h, {x, y});
}

```

```

void read(int pic[15][15], int numbersR[15][5], int numbersD[15][5], int
number){
    char c;
    int counter = 0;
    setlocale(LC_ALL, "eu");
    ifstream file ("file.txt");

    if (file.is_open()) {
        while(file){
            file >> c;
            if(c == '.'){
                if(counter == number){
                    for(int i = 0; i < 15; i++){
                        for(int j = 0; j < 15; j++){
                            file >> pic[i][j];
                        }
                    }
                    file >> c;
                    if(c == ','){
                        for(int i = 0; i < 15; i++){
                            for(int j = 0; j < 5; j++){
                                file >> numbersR[i][j];
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

                                for(int i = 0; i < 15; i++){
                                    for(int j = 0; j < 5; j++){
                                        file >> numbersD[i][j];
                                    }
                                    cout<<endl;
                                }
                                break;
                            }
                        }
                    counter++;
                }
            }
        }
    }
}

```

```

int main(){
    bool playing = false, escape = true, selectLvl = false, readingRules =
false;;
    int x = 2, y = 2;
    int step = 5;
    char c;
    int k = 0, n = 3;
    int pic[15][15], filled[15][15];
    int numbersR[15][5]; int numbersD[15][5];

    short status;

    HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD t = {0, 0}, b = {116, 52};

```

```

CONSOLE_SCREEN_BUFFER_INFO csbiInfo;
GetConsoleScreenBufferInfo(h, &csbiInfo);
SetConsoleTextAttribute(h, (WORD)((30 << 4) | 5));
system("cls");

```

```

SetConsoleCursorPosition(h, {x, y});

```

```

printMenu(h, {6, 3}, 0, 3, 20);
do{
    c = getch();
    switch(c){
case 75: //left
    if(x > 2 && playing){
        x -= step + 2;
        cursorMove(h, x, y, step, -1, 0);
    }
    else if(!playing){
        if(k == 0){
            k = n-1;
        }
        else
            k--;
        if(!selectLvl)
            printMenu(h, {6, 3}, k, n, 20);
        else
            printLvls(h, {6, 3}, k, n, 20);
    }
    break;
case 77: //right

```

```

        if(x < 100 && playing){
            x += step + 2;
            cursorMove(h, x, y, step, 1, 0);
        }
        else if(!playing){
            if(k == n-1){
                k = 0;
            }
            else
                k++;
            if(!selectLvl)
                printMenu(h, {6, 3}, k, n, 20);
            else
                printLvls(h, {6, 3}, k, n, 20);
        }
        break;
case 72: //down
    if(y > 3 && playing){
        y -= step - 2;
        cursorMove(h, x, y, step, 0, 1);
    }
    break;
case 80: //up
    if(y < 42 && playing){
        y += step - 2;
        cursorMove(h, x, y, step, 0, -1);
    }
    break;
case 32: //space

```

```

if(playing){
    playing = false;
    for(int i = 0; i < 15; i++){
        for(int j = 0; j < 15; j++){
            if(pic[i][j] == 1 && (filled[i][j] ==
0 || filled[i][j] == 2)){
                playing = true;
                break;
            }
        }
    }
    if(!playing){
        SetConsoleTextAttribute(h, (WORD)((0
<< 4) | 0));

        system("cls");
        k = 0;
        printMenu(h, {6, 3}, k, n, 20);
        break;
    }
    if(filled[(y-2)/3][(x-2)/7] == 0){
        if(pic[(y-2)/3][(x-2)/7] && playing){
            fill(h, x, y);
            filled[(y-2)/3][(x-2)/7] = 1;
        }
        else if(playing){
            cross(h, x, y, 20);
            filled[(y-2)/3][(x-2)/7] = 2;
        }
    }
}

```

```

    }
    break;
case 13: //enter
    if(!playing){
        for(int i = 0; i < 15; i++){
            for(int j = 0; j < 15; j++){
                filled[i][j] = 0;
            }
        }
        switch(k){
            case 0:
                printLvls(h, { 6, 3}, k, n, 20);
                if(selectLvl){
                    read(pic, numbersR,
numbersD, 0);

                    playing = true;
                    drawBond(h, t, b, numbersR,
numbersD);

                    x = 2; y = 2;
                }
                selectLvl = true;
                break;
            case 1:
                printRules(h);
                if(selectLvl){
                    read(pic, numbersR,
numbersD, 1);

                    playing = true;

```



```

numbersD);

drawBond(h, t, b, numbersR,

x = 2; y = 2;
}
readingRules = true;
break;
case 2:
    escape = false;
    break;
}
}
else{
    if(!filled[(y-2)/3][(x-2)/7]){
        if(pic[(y-2)/3][(x-2)/7]){
            cross(h, x, y, 20); // true
            filled[(y-2)/3][(x-2)/7] = 1;
        }
        else{
            cross(h, x, y, 24); // mimo
            filled[(y-2)/3][(x-2)/7] = 2;
        }
    }
}
break;
case 27: //escape
    playing = false; selectLvl = false; readingRules =
false;

SetConsoleTextAttribute(h, (WORD)((0 << 4) | 0));

```

```
        system("cls");  
        k = 0;  
        printMenu(h, {6, 3}, k, n, 20);  
        break;  
    }  
    }while(escape);  
    system("cls");  
}
```