

Курс «Основы программирования»

Власенко Олег Федосович

SimbirSoft

Лекция 10

Двухмерные массивы

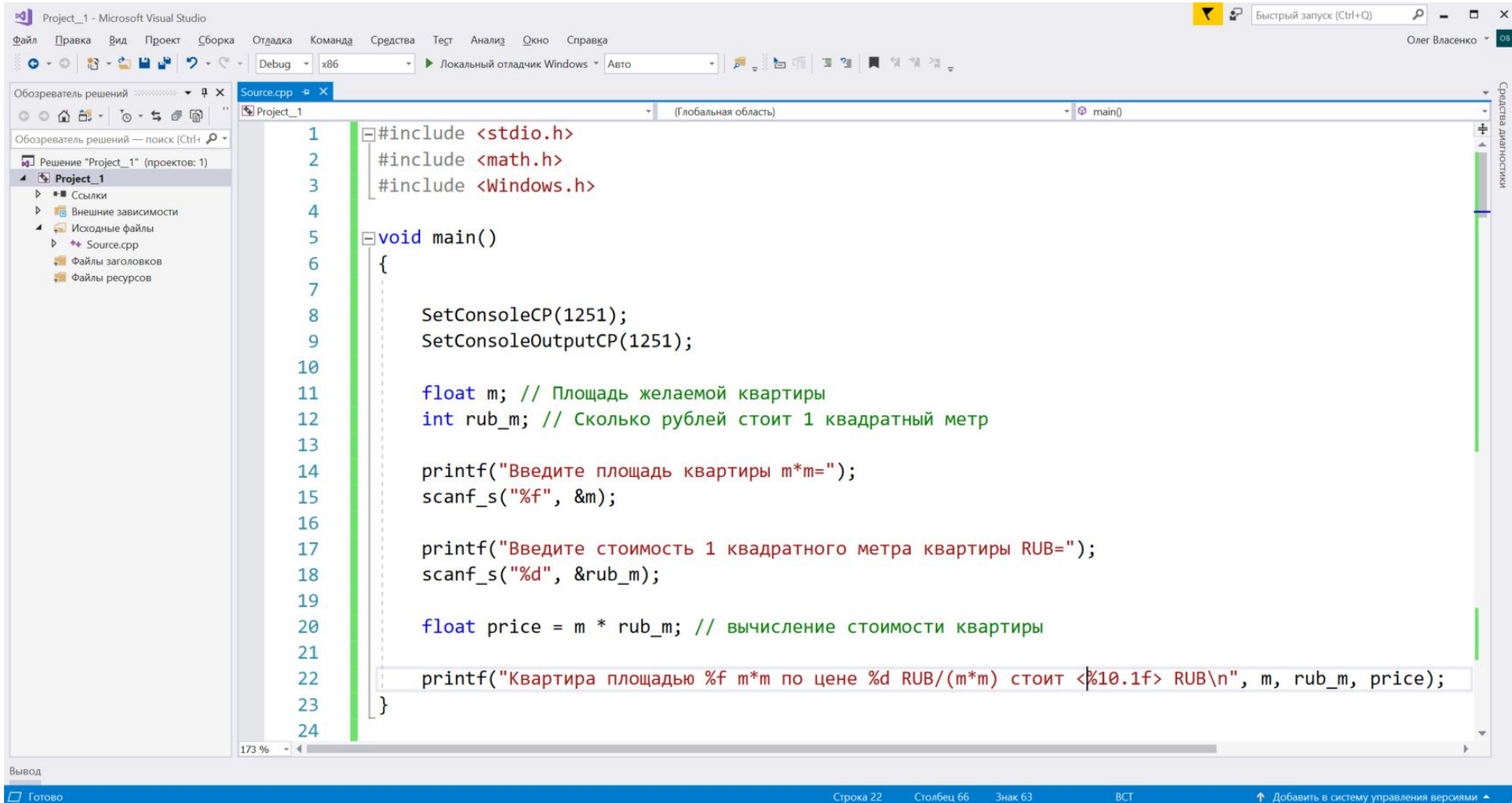
ЛР 18. Простейшие операции с двухмерными массивами

ЛР 19. Применение двухмерных массивов в играх

**Переменные, структуры,
массивы, массивы структур**

Лекция 2. «Покупаем квартиру»

Вводится площадь квартиры и стоимость квадратного метра.
Выводится сколько будет стоить эта квартира.



The screenshot shows the Microsoft Visual Studio IDE with a C++ source file named 'Source.cpp'. The code is as follows:

```
1 #include <stdio.h>
2 #include <math.h>
3 #include <Windows.h>
4
5 void main()
6 {
7
8     SetConsoleCP(1251);
9     SetConsoleOutputCP(1251);
10
11     float m; // Площадь желаемой квартиры
12     int rub_m; // Сколько рублей стоит 1 квадратный метр
13
14     printf("Введите площадь квартиры m*m=");
15     scanf_s("%f", &m);
16
17     printf("Введите стоимость 1 квадратного метра квартиры RUB=");
18     scanf_s("%d", &rub_m);
19
20     float price = m * rub_m; // вычисление стоимости квартиры
21
22     printf("Квартира площадью %f m*m по цене %d RUB/(m*m) стоит <%10.1f> RUB\n", m, rub_m, price);
23 }
24
```

The status bar at the bottom indicates 'Готово' (Ready), 'Строка 22' (Line 22), 'Столбец 66' (Column 66), 'Знак 63' (Character 63), 'ВСТ' (Encoding), and 'Добавить в систему управления версиями' (Add to version control).

Лекция 2. «Покупаем квартиру»

Вводится площадь квартиры и стоимость квадратного метра.
Выводится сколько будет стоить эта квартира.

```
1 #include <stdio.h>
2 #include <math.h>
3 #include <Windows.h>
4
5 void main()
6 {
7
8     SetConsoleCP(1251);
9     SetConsoleOutputCP(1251);
10
11     float m; // Площадь желаемой квартир
12     int rub_m; // Сколько рублей стоит 1
13
14     printf("Введите площадь квартиры m*m\n");
15     scanf_s("%f", &m);
16
17     printf("Введите стоимость 1 квадратн\n");
18     scanf_s("%d", &rub_m);
19
20     float price = m * rub_m; // вычислен
21
22     printf("Квартира площадью %f m*m по
23
24 }
```

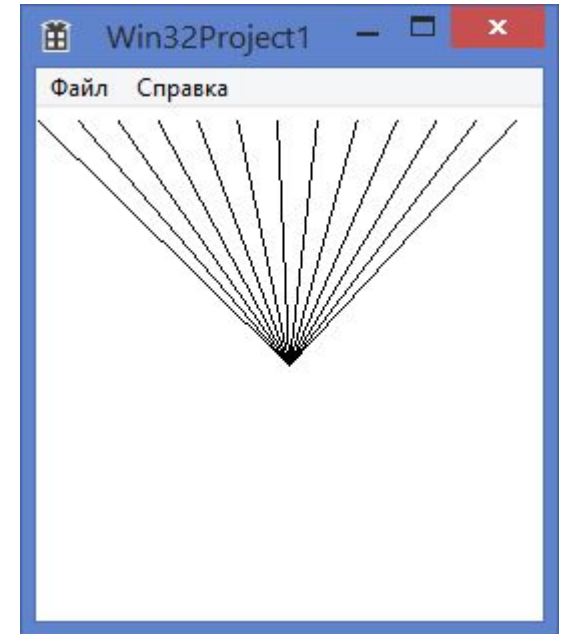
float	m	<input type="text"/>
int	rub_m	<input type="text"/>
float	price	<input type="text"/>

Лекция 7. «Рисуем много линии из центра»

```
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);

    RECT rect;
    GetClientRect(hWnd, &rect);

    int cx = rect.right / 2;
    int cy = rect.bottom / 2;
    int x = 0;
    while (x < rect.right) {
        MoveToEx(hdc, cx, cy, NULL);
        LineTo(hdc, x, 5);
        x += 20;
    }
    EndPaint(hWnd, &ps);
}
```



Лекция 7. «Рисуем много линии из центра»

```
case WM_PAINT:
```

```
{  
    PAINTSTRUCT ps;  
    HDC hdc = BeginPaint(hWnd, &ps);
```

```
    RECT rect;  
    GetClientRect(hWnd, &rect);
```

```
    int cx = rect.right / 2;  
    int cy = rect.bottom / 2;
```

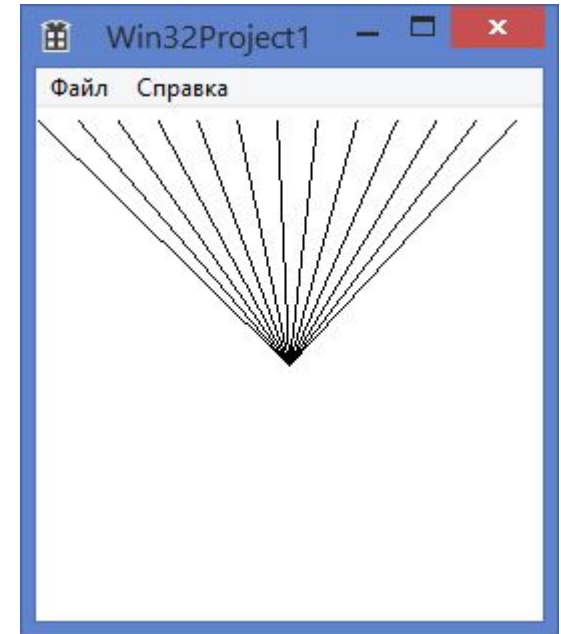
```
    int x = 0;
```

```
    while (x < rect.right) {  
        MoveToEx(hdc, cx, cy, NULL);  
        LineTo(hdc, x, 5);  
        x += 20;
```

```
    }
```

```
    EndPaint(hWnd, &ps);
```

```
}
```



<i>R</i>	<i>r</i>	<i>long</i>	left	
<i>E</i>	<i>e</i>	<i>long</i>	top	
<i>C</i>	<i>c</i>	<i>long</i>	right	
<i>T</i>	<i>t</i>	<i>long</i>	bottom	

Лекция 9. «Массив грибов»

В игре нужно собрать все грибы. В массивах MushroomX и MushroomY хранятся координаты каждого гриба. В массиве MushroomVisible хранится признак – «виден» / «не виден».

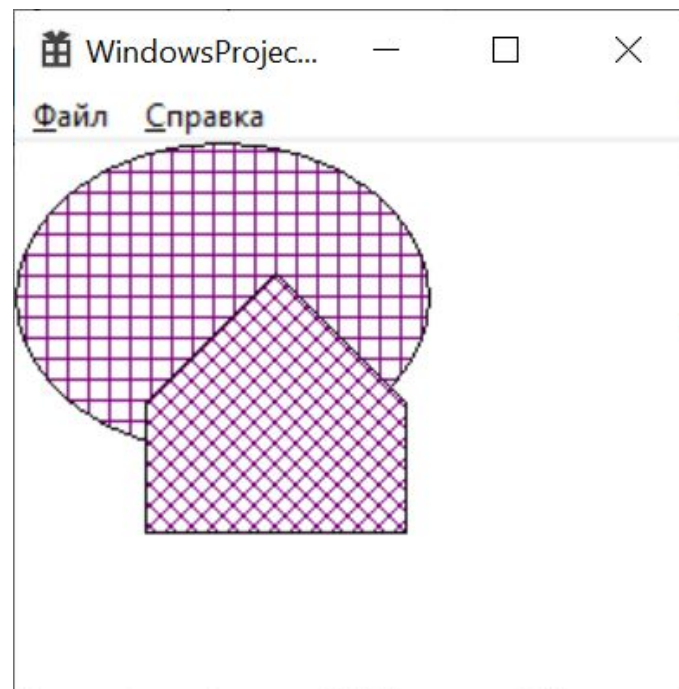
```
// Координаты грибов
// количество грибов
#define NUM_MUSHROOMS 12
// координата X каждого гриба
int MushroomX[NUM_MUSHROOMS] = { 200, 220, 240, 260,          300, 400, 420, 440,
                                460, 480, 500, 520 };
// координата Y каждого гриба
int MushroomY[NUM_MUSHROOMS] = { 400, 350, 300, 250,          300, 200, 100, 200,
                                100, 140, 120, 100 };
// виден ли гриб? (для каждого гриба!)
int MushroomVisible[NUM_MUSHROOMS] = { 1, 1, 1, 1,           1, 1, 1, 1,
                                        1, 1, 1, 1 };
```


Лекция 7 «Массивы&Структуры – пример – Polygon»

```
HBRUSH hBrush = CreateHatchBrush(HS_CROSS, RGB(128, 0, 128));  
SelectObject(hdc, hBrush);  
Ellipse(hdc, 0, 0, 160, 120);
```

```
HBRUSH hBrush2 = CreateHatchBrush(HS_DIAGCROSS, RGB(128, 0, 128));  
SelectObject(hdc, hBrush2);
```

```
POINT pt[5];  
pt[0].x = 100;  
pt[0].y = 50;  
pt[1].x = 150;  
pt[1].y = 100;  
pt[2].x = 150;  
pt[2].y = 150;  
pt[3].x = 50;  
pt[3].y = 150;  
pt[4].x = 50;  
pt[4].y = 100;  
Polygon(hdc, pt, 5);
```



Двухмерные массивы в Си

Двухмерные массивы – пример 1

```
#include <stdio.h>
```

```
void main() {
```

```
printf("main() start!\n");
```

```
int a[2][3];
```

```
a[0][0] = 1;
```

```
a[0][1] = 10;
```

```
a[0][2] = 100;
```

```
a[1][0] = 2;
```

```
a[1][1] = 20;
```

```
a[1][2] = 200;
```

```
printf("%d %d %d \n %d %d %d \n\n", a[0][0], a[0][1], a[0][2], a[1][0], a[1][1], a[1][2]);
```

```
int b[3][4] = { {1, 2, 3, 4}, {11, 12, 13, 14}, {21, 22, 23, 24} };
```

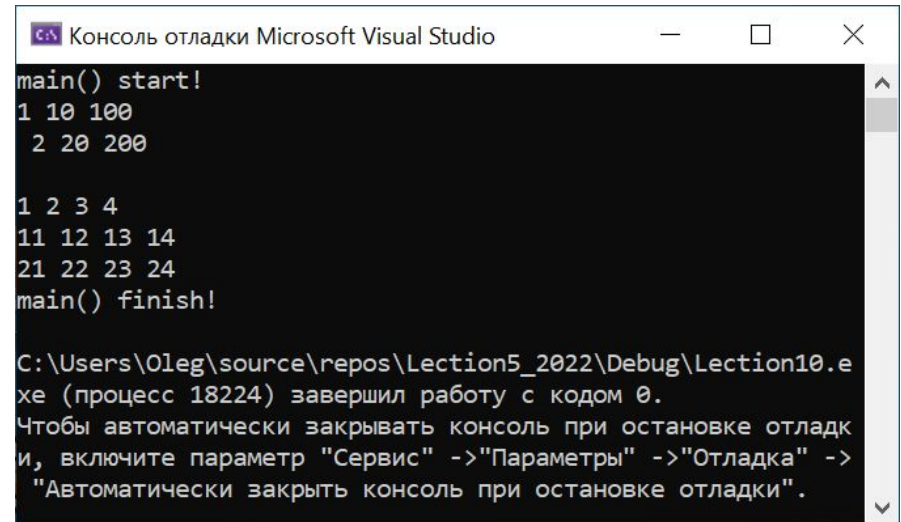
```
printf("%d %d %d %d \n", b[0][0], b[0][1], b[0][2], b[0][3]);
```

```
printf("%d %d %d %d \n", b[1][0], b[1][1], b[1][2], b[1][3]);
```

```
printf("%d %d %d %d \n", b[2][0], b[2][1], b[2][2], b[2][3]);
```

```
printf("main() finish!\n");
```

```
}
```



```
Консоль отладки Microsoft Visual Studio
main() start!
1 10 100
2 20 200

1 2 3 4
11 12 13 14
21 22 23 24
main() finish!

C:\Users\Oleg\source\repos\Lecton5_2022\Debug\Lecton10.exe (процесс 18224) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" ->"Автоматически закрывать консоль при остановке отладки".
```

Двухмерные массивы – пример 1

```
#include <stdio.h>

void main() {

printf("main() start!\n");

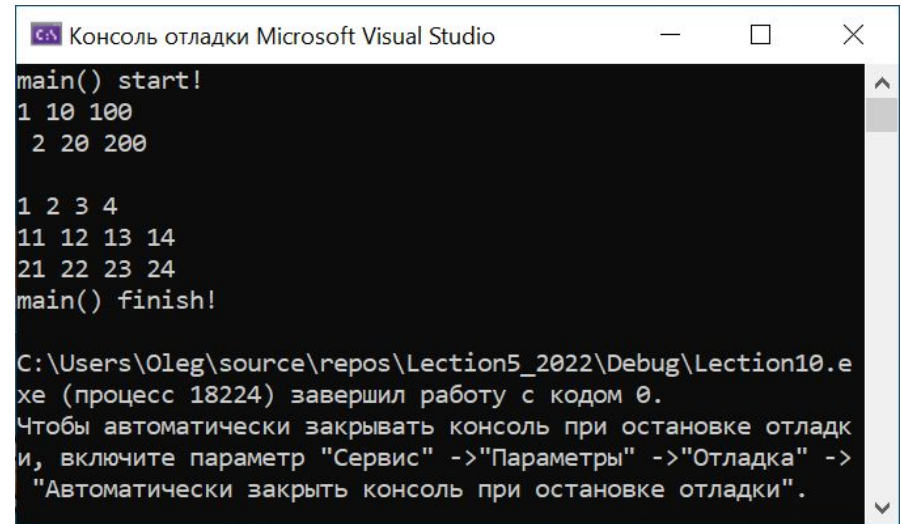
int a[2][3];
a[0][0] = 1;
a[0][1] = 10;
a[0][2] = 100;
a[1][0] = 2;
a[1][1] = 20;
a[1][2] = 200;

printf("%d %d %d \n %d %d %d \n\n", a[0][0], a[0][1], a[0][2], a[1][0], a[1][1], a[1][2]);

int b[3][4] = { {1, 2, 3, 4}, {11, 12, 13, 14}, {21, 22, 23, 24} };

printf("%d %d %d %d \n", b[0][0], b[0][1], b[0][2], b[0][3]);
printf("%d %d %d %d \n", b[1][0], b[1][1], b[1][2], b[1][3]);
printf("%d %d %d %d \n", b[2][0], b[2][1], b[2][2], b[2][3]);

printf("main() finish!\n");
}
```



```
Консоль отладки Microsoft Visual Studio
main() start!
1 10 100
2 20 200

1 2 3 4
11 12 13 14
21 22 23 24
main() finish!

C:\Users\Oleg\source\repos\Lecton5_2022\Debug\Lecton10.exe (процесс 18224) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" ->"Автоматически закрыть консоль при остановке отладки".
```

int	a	[0]	[1]	[2]	[3]
[0]		1	10	100	
[1]		2	20	200	
[2]					

int	b	[0]	[1]	[2]	[3]	[4]
[0]		1	2	3	4	
[1]		11	12	13	14	
[2]		21	22	23	24	
[3]						

Двухмерные массивы – пример 1

```
int a[2][3];  
a[0][0] = 1;  
a[0][1] = 10;  
a[0][2] = 100;  
a[1][0] = 2;  
a[1][1] = 20;  
a[1][2] = 200;
```

<i>int</i>	a	[0]	[1]	[2]	[3]
	[0]	1	10	100	
	[1]	2	20	200	
	[2]				

```
int b[3][4] = { {1, 2, 3, 4}, {11, 12, 13, 14}, {21, 22, 23, 24} };
```

<i>int</i>	b	[0]	[1]	[2]	[3]	[4]
	[0]	1	2	3	4	
	[1]	11	12	13	14	
	[2]	21	22	23	24	
	[3]					

Двухмерные массивы – пример 2

```
#include <stdio.h>

void main() {
    printf("main() start!\n");
```

```
    int a[2][3];
    a[0][0] = 1;
    a[0][1] = 10;
    a[0][2] = 100;
    a[1][0] = 2;
    a[1][1] = 20;
    a[1][2] = 200;
```

```
    int b[3][4] = { {1, 2, 3, 4}, {11, 12, 13, 14}, {21, 22, 23, 24} };
```

```
    int i, j;
```

```
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 3; j++) {
            printf("%d", a[i][j]);
        }
    }
```

```
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 4; j++) {
            printf("%d", b[i][j]);
        }
    }
```

```
    printf("main() finish!\n");
}
```

C:\> Выбрать Консоль отладки Microsoft Visual Studio

```
main() start!
11010022020012341112131421222324main() finish!
```

Двухмерные массивы – пример 3

```
#include <stdio.h>

void main() {
    printf("main() start!\n");
```

```
    int a[2][3];
    a[0][0] = 1;
    a[0][1] = 10;
    a[0][2] = 100;
    a[1][0] = 2;
    a[1][1] = 20;
    a[1][2] = 200;
```

```
    int b[3][4] = { {1, 2, 3, 4}, {11, 12, 13, 14}, {21, 22, 23, 24} };
```

```
    int i, j;
```

```
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 3; j++) {
            printf("%d ", a[i][j]);
        }
    }
```

```
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 4; j++) {
            printf("%d ", b[i][j]);
        }
    }
```

```
    printf("main() finish!\n");
}
```

Консоль отладки Microsoft Visual Studio

```
main() start!
1 10 100 2 20 200 1 2 3 4 11 12 13 14 21 22 23 24 main() finish!
```

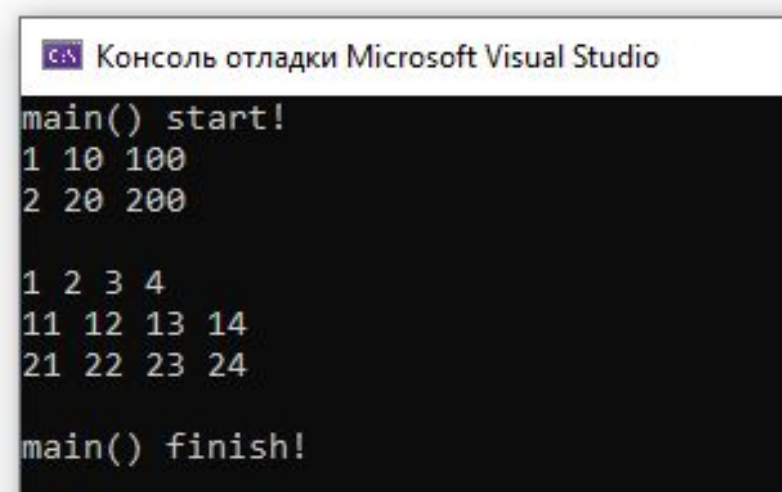
Двухмерные массивы – пример 4

```
#include <stdio.h>

void main() {
printf("main() start!\n");

int a[2][3];
a[0][0] = 1;a[0][1] = 10;a[0][2] = 100;
a[1][0] = 2;a[1][1] = 20;a[1][2] = 200;
int b[3][4] = { {1, 2, 3, 4}, {11, 12, 13, 14}, {21, 22, 23, 24} };

for (i = 0; i < 2; i++) {
    for (j = 0; j < 3; j++) {
        printf("%d ", a[i][j]);
    }
    printf("\n");
}
printf("\n");
for (i = 0; i < 3; i++) {
    for (j = 0; j < 4; j++) {
        printf("%d ", b[i][j]);
    }
    printf("\n");
}
printf("\n");
printf("main() finish!\n");
}
```



Консоль отладки Microsoft Visual Studio

```
main() start!
1 10 100
2 20 200

1 2 3 4
11 12 13 14
21 22 23 24

main() finish!
```

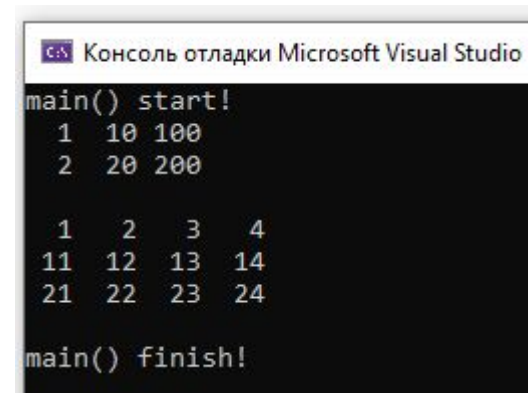
Двухмерные массивы – пример 5

```
#include <stdio.h>

void main() {
printf("main() start!\n");

int a[2][3];
a[0][0] = 1;a[0][1] = 10;a[0][2] = 100;
a[1][0] = 2;a[1][1] = 20;a[1][2] = 200;
int b[3][4] = { {1, 2, 3, 4}, {11, 12, 13, 14}, {21, 22, 23, 24} };

for (i = 0; i < 2; i++) {
    for (j = 0; j < 3; j++) {
        printf("%3d ", a[i][j]);
    }
    printf("\n");
}
printf("\n");
for (i = 0; i < 3; i++) {
    for (j = 0; j < 4; j++) {
        printf("%3d ", b[i][j]);
    }
    printf("\n");
}
printf("\n");
printf("main() finish!\n");
}
```



```
Консоль отладки Microsoft Visual Studio
main() start!
 1  10 100
 2  20 200

 1  2  3  4
11 12 13 14
21 22 23 24

main() finish!
```


Двухмерный массив в Си

Некоторые операции

Вывод элементов массива

```
#include <stdio.h>
#include <Windows.h>
```

```
#define MAX_N 8
#define MAX_M 10
```

```
int arr[MAX_N][MAX_M] = {
    { 0, 1, 222, 3},
    {10, 11, 12, 13},
    {20, 21, 22, 23}
```

```
};
int n = 3;
int m = 4;
```

```
void print() {
    printf("!!!! print() !!!!\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            printf("%3d ", arr[i][j]);
        }
        printf("\n");
    }
}
```

```
void main() {
    print();
}
```

						m = 4							
int	arr	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
	[0]	0	1	222	3								
	[1]	10	11	12	13								
	[2]	20	21	22	23								
n = 3	[3]												
	[4]												
	[5]												
	[6]												
	[7]												
	[8]												
	[9]												
	[10]												
	[11]												

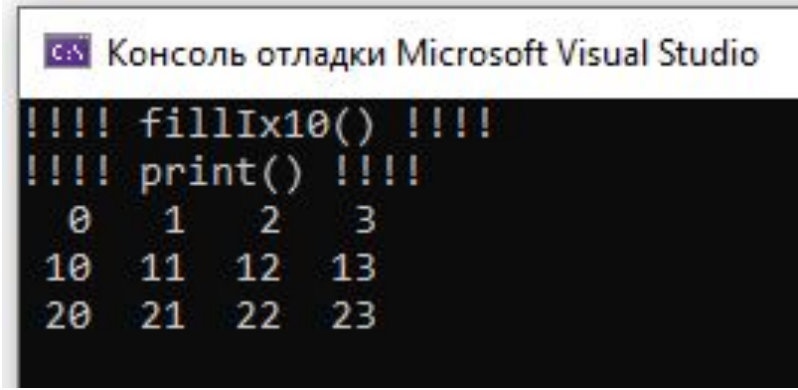
Консоль отладки Microsoft Visual Studio

```
!!!! print() !!!!
  0  1 222  3
10 11 12 13
20 21 22 23
```

Заполнение значениями $i * 10 + j$

```
void fillIx10() {  
    printf("!!!! fillIx10() !!!!\n");  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < m; j++) {  
            arr[i][j] = i * 10 + j;  
        }  
    }  
}
```

```
void main() {  
    SetConsoleCP(1251);  
    SetConsoleOutputCP(1251);  
    fillIx10();  
    print();  
}
```



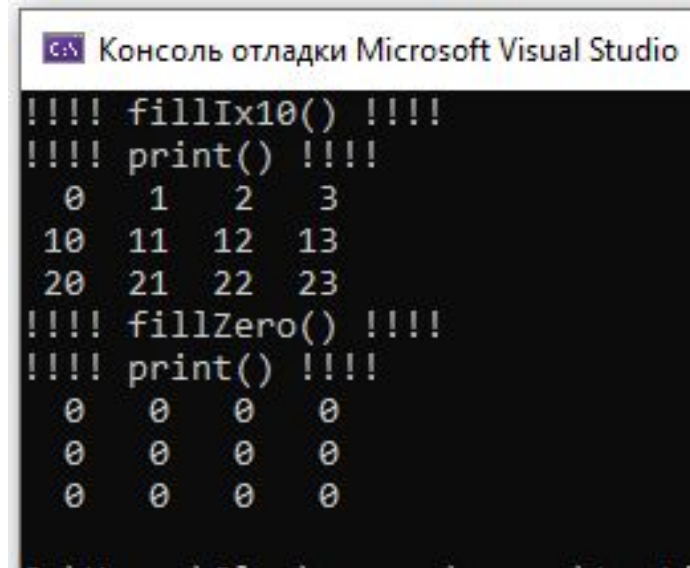
Консоль отладки Microsoft Visual Studio

```
!!!! fillIx10() !!!!  
!!!! print() !!!!  
 0  1  2  3  
10 11 12 13  
20 21 22 23
```


Заполнение значениями 0

```
void fillZero() {  
    printf("!!!! fillZero() !!!!\n");  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < m; j++) {  
            arr[i][j] = 0;  
        }  
    }  
}
```

```
void main() {  
    SetConsoleCP(1251);  
    SetConsoleOutputCP(1251);  
  
    fillIx10();  
    print();  
    fillZero();  
    print();  
}
```

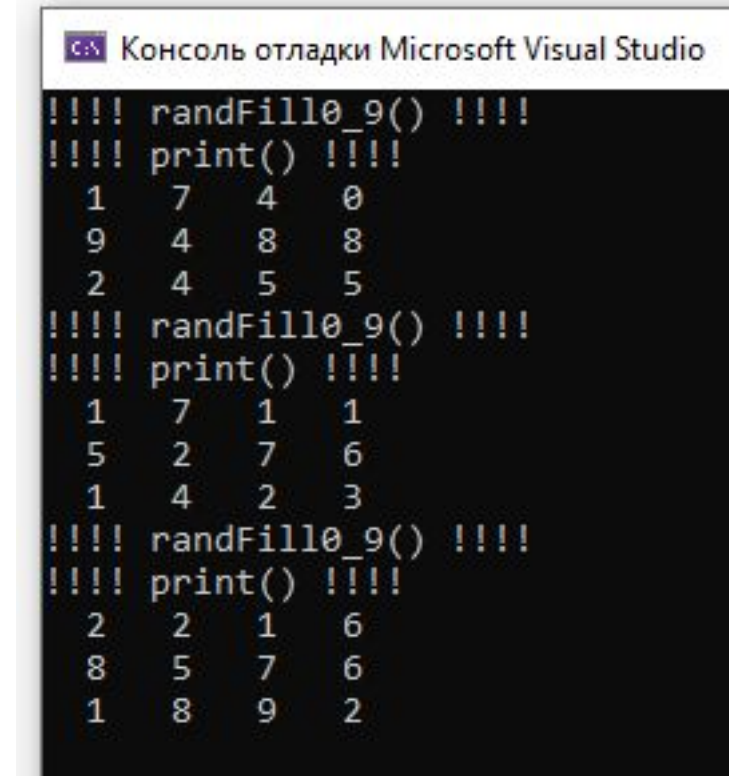


```
Консоль отладки Microsoft Visual Studio  
!!!! fillIx10() !!!!  
!!!! print() !!!!  
0 1 2 3  
10 11 12 13  
20 21 22 23  
!!!! fillZero() !!!!  
!!!! print() !!!!  
0 0 0 0  
0 0 0 0  
0 0 0 0
```

Заполнение случайными значениями

```
void randFill0_9() {  
    printf("!!!! randFill0_9() !!!!\n");  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < m; j++) {  
            arr[i][j] = rand() % 10;  
        }  
    }  
}
```

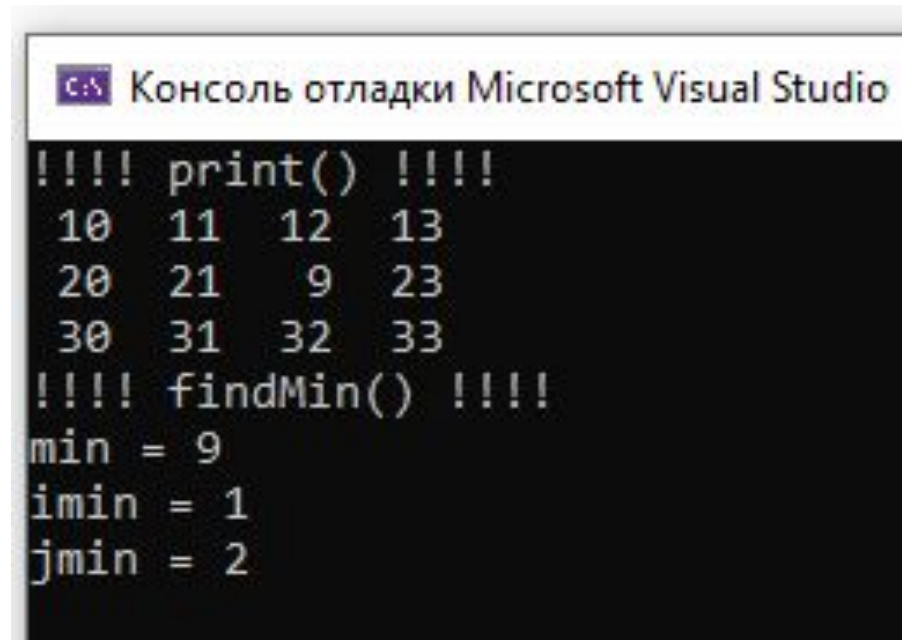
```
void main() {  
    SetConsoleCP(1251);  
    SetConsoleOutputCP(1251);  
  
    randFill0_9();  
    print();  
    randFill0_9();  
    print();  
    randFill0_9();  
    print();  
}
```



```
Консоль отладки Microsoft Visual Studio  
!!!! randFill0_9() !!!!  
!!!! print() !!!!  
1 7 4 0  
9 4 8 8  
2 4 5 5  
!!!! randFill0_9() !!!!  
!!!! print() !!!!  
1 7 1 1  
5 2 7 6  
1 4 2 3  
!!!! randFill0_9() !!!!  
!!!! print() !!!!  
2 2 1 6  
8 5 7 6  
1 8 9 2
```

Поиск минимального элемента

```
void findMin() {  
    printf("!!!! findMin() !!!!\n");  
    int min = arr[0][0];  
    int iMin = 0;  
    int jMin = 0;  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < m; j++) {  
            if (arr[i][j] < min) {  
                min = arr[i][j];  
                iMin = i;  
                jMin = j;  
            }  
        }  
    }  
    printf("min = %d\n", min);  
    printf("imin = %d\n", iMin);  
    printf("jmin = %d\n", jMin);  
}  
  
void main() {  
    print();  
    findMin();  
}
```



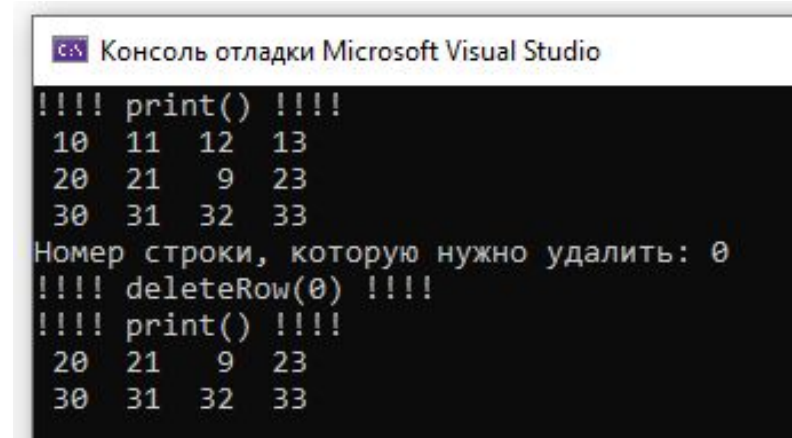
С:\> Консоль отладки Microsoft Visual Studio

```
!!!! print() !!!!  
10 11 12 13  
20 21 9 23  
30 31 32 33  
!!!! findMin() !!!!  
min = 9  
imin = 1  
jmin = 2
```

Удалить строку

```
void deleteRow(int delRow) {  
    printf("!!!! deleteRow(%d) !!!!\n", delRow);  
    for (int i = delRow; i < n - 1; i++) {  
        for (int j = 0; j < m; j++) {  
            arr[i][j] = arr[i + 1][j];  
        }  
    }  
    n--;  
}
```

```
void main() {  
    SetConsoleCP(1251);  
    SetConsoleOutputCP(1251);  
    print();  
    int num;  
    printf("Номер строки, которую нужно удалить: ");  
    scanf_s("%d", &num);  
    deleteRow(num);  
    print();  
}
```



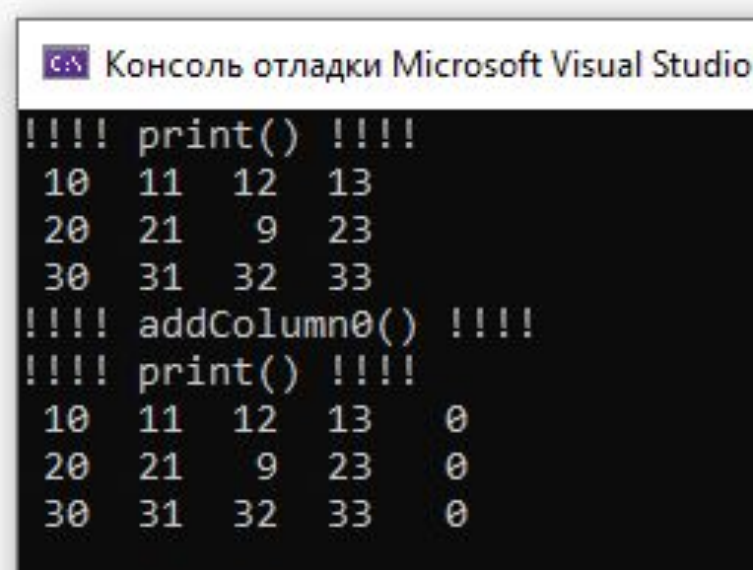
Консоль отладки Microsoft Visual Studio

```
!!!! print() !!!!  
10 11 12 13  
20 21 9 23  
30 31 32 33  
Номер строки, которую нужно удалить: 0  
!!!! deleteRow(0) !!!!  
!!!! print() !!!!  
20 21 9 23  
30 31 32 33
```

Вставить столбец

```
void addColumn0() {  
    printf("!!!! addColumn0() !!!!\n");  
  
    if (m < MAX_M) {  
        for (int i = 0; i < n; i++) {  
            arr[i][m] = 0;  
        }  
        m++;  
    }  
}
```

```
void main() {  
    SetConsoleCP(1251);  
    SetConsoleOutputCP(1251);  
  
    print();  
    addColumn0();  
    print();  
}
```



Консоль отладки Microsoft Visual Studio

```
!!!! print() !!!!  
10  11  12  13  
20  21   9  23  
30  31  32  33  
!!!! addColumn0() !!!!  
!!!! print() !!!!  
10  11  12  13  0  
20  21   9  23  0  
30  31  32  33  0
```


Лабораторная работа №18

Простейшие операции с двумерными массивами

Задача 1. Создать консольное меню

Создать меню, при помощи которого можно выбирать операции над двумерным массивом. Выводить состояние массива перед каждым обращением к меню.

```
C:\Users\Oleg\source\repos\Labs\Debug\Lab18.exe

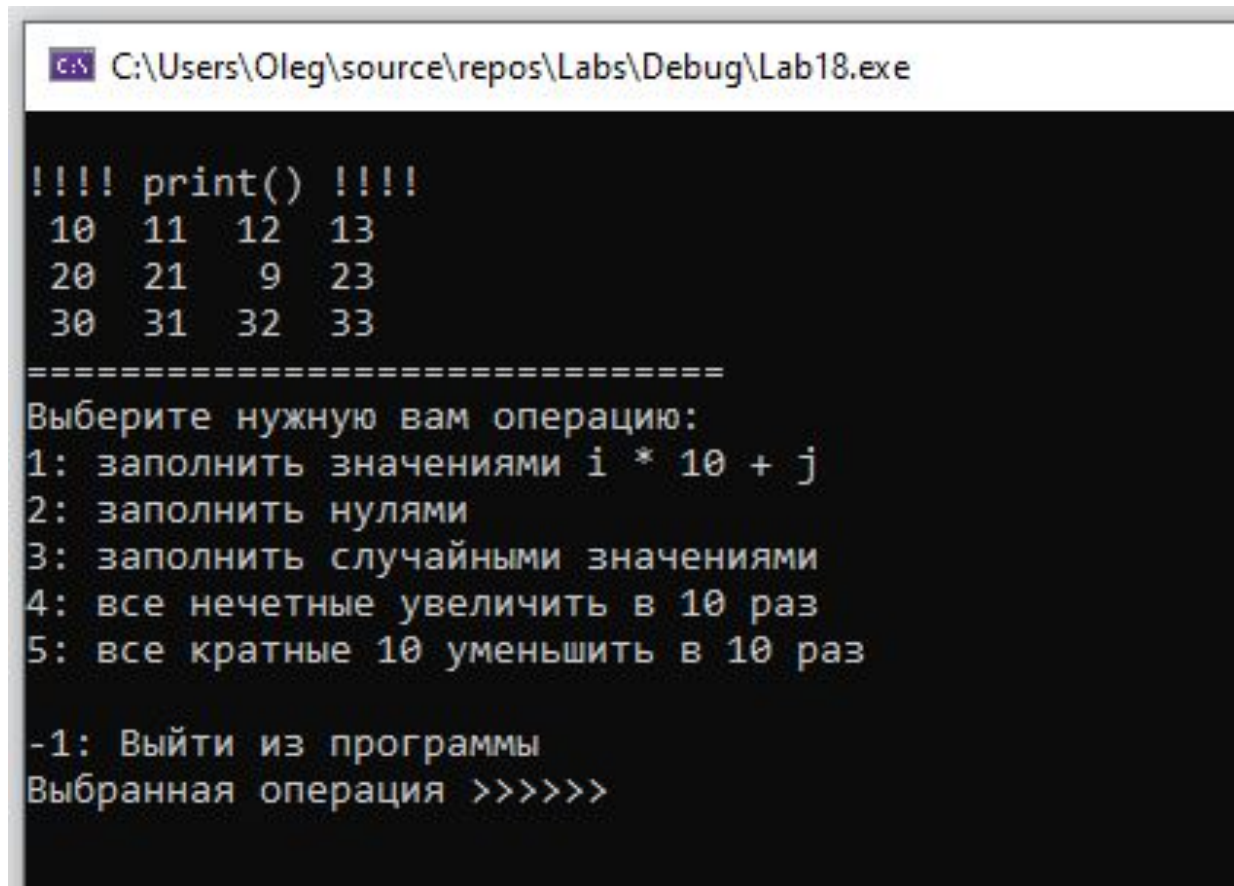
!!!! print() !!!!
10 11 12 13
20 21  9 23
30 31 32 33
=====
Выберите нужную вам операцию:
1: заполнить значениями  $i * 10 + j$ 
2: заполнить нулями
3: заполнить случайными значениями
4: все нечетные увеличить в 10 раз
5: все кратные 10 уменьшить в 10 раз

-1: Выйти из программы
Выбранная операция >>>>>
```


Задача 2. Реализовать операции

Прикрутить операции, ранее разобранные в лекции:

1. Заполнить значениями $i * 10 + j$
2. Заполнить нулями
3. Заполнить случайными значениями



```
C:\Users\Oleg\source\repos\Labs\Debug\Lab18.exe

!!!! print() !!!!
10 11 12 13
20 21  9 23
30 31 32 33
=====
Выберите нужную вам операцию:
1: заполнить значениями  $i * 10 + j$ 
2: заполнить нулями
3: заполнить случайными значениями
4: все нечетные увеличить в 10 раз
5: все кратные 10 уменьшить в 10 раз

-1: Выйти из программы
Выбранная операция >>>>>
```

Задача 3. Реализовать операцию

Реализовать и прикрутить к меню операцию – «Все нечетные увеличить в 10 раз»

```
C:\Users\Oleg\source\repos\Labs\Debug\Lab18.exe

!!!! print() !!!!
10 11 12 13
20 21  9 23
30 31 32 33
=====
Выберите нужную вам операцию:
1: заполнить значениями  $i * 10 + j$ 
2: заполнить нулями
3: заполнить случайными значениями
4: все нечетные увеличить в 10 раз
5: все кратные 10 уменьшить в 10 раз

-1: Выйти из программы
Выбранная операция >>>>>
```

Задача 4. Реализовать операцию

Реализовать и прикрутить к меню операцию – «Все кратные 10 уменьшить в 10 раз»

```
C:\Users\Oleg\source\repos\Labs\Debug\Lab18.exe

!!!! print() !!!!
10 11 12 13
20 21  9 23
30 31 32 33
=====
Выберите нужную вам операцию:
1: заполнить значениями  $i * 10 + j$ 
2: заполнить нулями
3: заполнить случайными значениями
4: все нечетные увеличить в 10 раз
5: все кратные 10 уменьшить в 10 раз

-1: Выйти из программы
Выбранная операция >>>>>
```

Домашнее задание по ЛР18

- 1) Доделать задачи 1-4.
- 2) Добавить в программу возможность ввода массива с клавиатуры
- 3) * удалить заданную строку из массива
- 4) * вставить в конец массива столбец, содержащий нули
- 5) * элементы стоящие левее максимального увеличить в 2 раза
- 5) * элементы стоящие ниже максимального увеличить в 5 раз

ИТОГО по ЛР18

1. Познакомились с несколькими операциями над двумерными массивами

Использование двухмерного массива для игры

Как представить карту

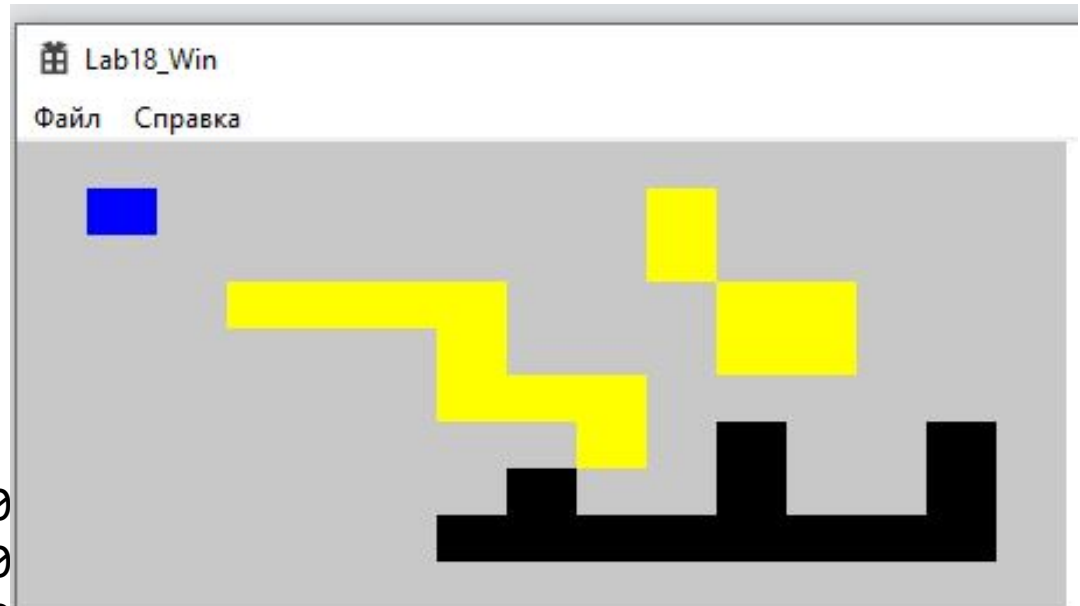
```
#define N 10
#define M 15
// Коды ячеек:
// 0 - свободна
// 1 - игрок
// 2 - препятствие
// 3 - золото
int map[N][M] = {
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0},
    {0, 0, 0, 3, 3, 3, 3, 0, 0, 0, 3, 3, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 3, 3, 0, 0, 0},

    {0, 0, 0, 0, 0, 0, 3, 3, 3, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 2, 0, 0, 2, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0},
    {0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
};
```

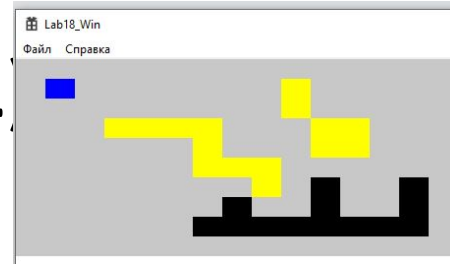

Как представить карту

```
#define N 10
#define M 15
// Коды ячеек:
// 0 - свободна
// 1 - игрок
// 2 - препятствие
// 3 - золото
int map[N][M] = {
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 3, 3, 3, 3, 0, 0, 0, 3, 3, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 0, 3, 3, 0, 0},

    {0, 0, 0, 0, 0, 0, 3, 3, 3, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 2, 0, 0, 2},
    {0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2, 0, 0, 2},
    {0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
};
```



Как нарисовать карту (1)



```
#define WIDTH 30
#define HEIGHT 20

void drawMap(HDC hdc) {
    // кисть для пустого поля
    HBRUSH hBrushEmptyCell = CreateSolidBrush(RGB(200, 200, 200)); // серый
    // кисть для поля с золотом
    HBRUSH hBrushGold = CreateSolidBrush(RGB(255, 255, 0)); // желтый
    // кисть для стены
    HBRUSH hBrushWall = CreateSolidBrush(RGB(0, 0, 0)); // черный
    // кисть для игрока
    HBRUSH hBrushMan = CreateSolidBrush(RGB(0, 0, 255)); // синий

    // Коды ячеек:
    // 0 - свободна
    // 1 - игрок
    // 2 - препятствие
    // 3 - золото
    HBRUSH brush[4] = { hBrushEmptyCell, hBrushMan, hBrushWall, hBrushGold };
```

Как нарисовать карту (2)

```
int i, j;

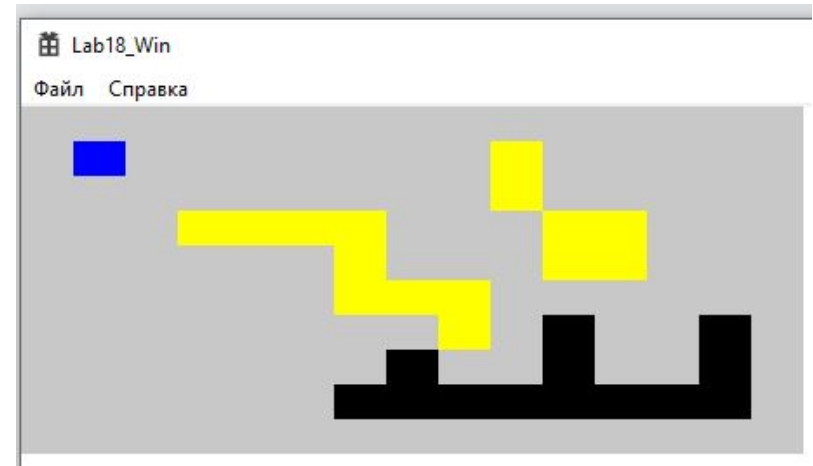
for (i = 0; i < N; i++) {
    for (j = 0; j < M; j++) {

        int x1 = j * WIDTH;
        int x2 = (j + 1) * WIDTH;
        int y1 = i * HEIGHT;
        int y2 = (i + 1) * HEIGHT;

        RECT r = { x1, y1, x2, y2 };
        FillRect(hdc, &r, brush[map[i][j]]);
    }
}
```

```
// Все кисти удаляем!!!
for (i = 0; i < 4; i++)
    DeleteObject(brush[i]);
```

```
}
```



Как героев подвинуть влево

```
int steps = 0;  
int gold = 0;
```

```
void Left() {  
    int i, j;
```

```
    for (i = 0; i < N; i++) {  
        for (j = 1; j < M; j++) {  
            if (map[i][j] == 1) { // если в map[i][j] игрок  
                if (map[i][j - 1] == 0) { // если слева от игрока - пустая клетка  
                    map[i][j - 1] = 1;  
                    map[i][j] = 0;  
                    steps++;  
                }  
                else if (map[i][j - 1] == 3) { // если слева от игрока - золото  
                    map[i][j - 1] = 1;  
                    map[i][j] = 0;  
                    steps++;  
                    gold++;  
                }  
            }  
        }  
    }  
}
```

Как героев подвинуть вправо

```
void Right() {
    int i, j;

    for (i = 0; i < N; i++) {

        for (j = M - 2; j >= 0; j--) {

            if (map[i][j] == 1) { // если в map[i][j] игрок
                if (map[i][j + 1] == 0) {
                    map[i][j + 1] = 1;
                    map[i][j] = 0;
                    steps++;
                }
                else if (map[i][j + 1] == 3) {
                    map[i][j + 1] = 1;
                    map[i][j] = 0;
                    steps++;
                    gold++;
                }
            }
        }
    }
}
```

Как героев подвинуть вверх

```
void Up() {
    int i, j;

    for (i = 1; i < N; i++) {

        for (j = 0; j < M; j++) {

            if (map[i][j] == 1) { // если в map[i][j] игрок
                if (map[i - 1][j] == 0) {
                    map[i - 1][j] = 1;
                    map[i][j] = 0;
                    steps++;
                }
                else if (map[i - 1][j] == 3) {
                    map[i - 1][j] = 1;
                    map[i][j] = 0;
                    steps++;
                    gold++;
                }
            }
        }
    }
}
```

Как управлять перемещением героев

```
case WM_KEYDOWN:
    switch (wParam)
    {
    case VK_DOWN:
        Down();
        InvalidateRect(hWnd, NULL, TRUE);
        break;
    case VK_LEFT:
        Left();
        InvalidateRect(hWnd, NULL, TRUE);
        break;
    case VK_UP:
        Up();
        InvalidateRect(hWnd, NULL, TRUE);
        break;
    case VK_RIGHT:
        Right();
        InvalidateRect(hWnd, NULL, TRUE);
        break;
    }
    break;
```

Как отрисовывать карту

```
case WM_PAINT:  
    {  
        PAINTSTRUCT ps;  
        HDC hdc = BeginPaint(hWnd, &ps);  
        // TODO: Добавьте сюда любой код прорисовки, использующий HDC...  
  
        drawMap(hdc);  
  
        EndPaint(hWnd, &ps);  
    }  
    break;
```


Лабораторная работа №19

Применение двумерных массивов в
играх

Задача 1. Собрать игру из кода

Из кода, предоставленного выше в данной лекции нужно собрать заготовку игры.

Задача 2. Доделать управление

В коде выше нет перемещения героев вниз. Нужно создать код, который будет обеспечивать перемещение героев вниз!

Задача 3*.

При нажатии клавиши 'L' слева от игрока вставить элемент стены.

Задача 4*.

При нажатии клавиши 'R' непосредственно справа от игрока вставить элемент стены.

Домашнее задание

- Доделать задачи 1-4

- Задача 5

Придумать что-то на нажатие клавиши U (Up – «вверх») и реализовать это (стена/ золото создать/разрушить) для клетки выше игрока

- Задача 6.

Придумать что-то на нажатие клавиши D (Down – «вниз») и реализовать это (стена/ золото создать/разрушить) для клетки ниже игрока

ИТОГО по лекции 10

1. Разобрались с рядом операций над двухмерными массивами
2. Узнали что нужно сделать в ЛР18 и ЛР19