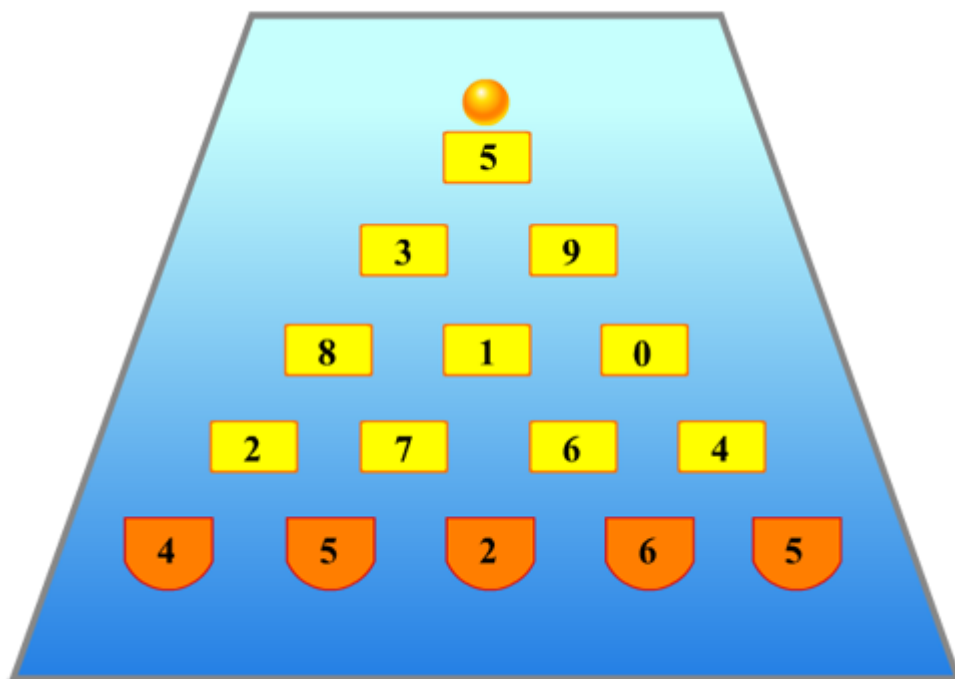


## Задание

Разработайте модель игры «Ступеньки».

С верхней ступеньки скатывается шарик. За один шаг шарик переходит только на одну ступеньку вниз по диагонали вправо или влево. По ходу движения шарик «собирает» очки, обозначенные на ступеньках. Очки суммируются. В какую из луз, находящихся в основании треугольника ступенек, должен попасть шарик, чтобы сумма очков была максимальной? Числа на ступеньках расставляются случайным образом. При новом запуске шарика расстановка чисел меняется.



Обозначим  $N$  – количество уровней ступенек, включая лузы. В нашем примере  $N = 5$ .

Представим исходные данные в виде таблицы  $A$  из 5 строк и 5 столбцов:

5				
3	9			
8	1	0		
2	7	6	4	
4	5	2	6	5

В программе таблица может быть представлена в виде двумерного массива  $A(5,5)$ . Индексы элементов массива будут принимать значения от 0 до 4.

Создадим таблицу  $B$  результатов движения шарика на каждом уровне, выбирая варианты движения, приносящие максимальное количество очков.

Например, попасть на ступеньку «1» на третьем уровне можно пройдя ступеньки:

- «5», «3» и «1», набрав очки  $5 + 3 + 1 = 9$ ;
- «5», «9» и «1», набрав очки  $5 + 9 + 1 = 15$ .

По условию игры нам нужна максимальная сумма, поэтому в таблицу занесем 15. Продолжая далее, получим таблицу максимальных очков, собранных при движении шарика на каждом уровне:

0	5				
0	8	14			
0	16	15	14		
0	18	23	21	18	
0	22	28	25	27	23




В таблицу  $B$  введен дополнительный столбец. Для того чтобы получать сумму на следующем уровне, необходимо складывать сумму очков «соседей» слева и справа на предыдущем уровне. Для первого столбца таблицы  $A$  «соседей» слева нет. Чтобы при разработке программы не выйти за границы индексов двумерного массива, объявим  $B(N, N + 1)$ .

В общем виде элементы таблицы  $B$ , кроме элементов нулевой строки и нулевого столбца, найдем по формуле:  $B_{ij} = \max(A_{i(j-1)} + B_{(i-1)j}, A_{i(j-1)} + B_{(i-1)(j-1)})$ , где  $\max$  – функция нахождения максимального элемента из элементов указанных в скобках (входит в набор стандартных функций в системе «1С:Предприятие»).

Программа будет включать:

- объявление и ввод переменной  $N$  – количество уровней «ступенек», определяющих размерность массивов;
- объявление массивов  $A(N, N)$  и  $B(N; N + 1)$ ;
- заполнение массивов нулями;
- заполнение элементов массива  $A$ , лежащих ниже главной диагонали, случайными числами. Вывод массива  $A$ ;
- заполнение массива  $B$ :  $B_{01} = A_{00}$ , остальные элементы по правилу  $B_{ij} = \max(A_{ij} + B_{(i-1)j}, A_{ij} + B_{(i-1)(j-1)})$ . Вывод массива  $B$ ;
- поиск максимального элемента в последней строке массива  $B$ . Вывод максимального элемента и его номера.

Придем к разработке программы на языке «1С:Предприятие»:

1. Откроем информационную базу *Мой класс* в режиме *Конфигуратор*.
2. Выделим в дереве конфигурации Обработки –  *Добавить (Ins)*.
3. В окне разработки на вкладке *Основные* введем имя *ИграСтупеньки*.
4. На вкладке *Формы* –  *Добавить (Ins)* – ОК.
5. В окне разработки формы на вкладке *Команды* –  *Добавить (Ins)*, имя *НайтиКорень*.
6. Перейдем в модуль формы, начнем вводить программный код.

В начале программы объявим переменную  $N$  – количество уровней «ступенек» – и введем ее:

```
НаКлиенте
Процедура НоваяИгра(Команда)
    Переменная N;
    ВвестиЧисло (N, "Введите количество уровней", 0);
```

Создадим с помощью конструктора Новый **двумерные массивы**:  $A$  – массив исходных случайных чисел,  $B$  – массив максимальных сумм:

```
A = Новый Массив (N,N);
B = Новый Массив (N,N+1);
```

**Заполним массивы** нулями с помощью специально написанной функции:

```
A = ЗаполнениеМассива (A,N,N);
B = ЗаполнениеМассива (B,N,N+1);
```

```
Функция ЗаполнениеМассива (Массив, K, L)
    Для i=0 По K-1 Цикл
        Для j=0 По L-1 Цикл
            Массив[i][j] = 0;
        КонецЦикла;
    КонецЦикла;
    Возврат (Массив)
КонецФункции
```

Заполним массив  $A$  случайными числами ниже главной диагонали и **выведем массив** в окно сообщений:

```
ГСЧ = Новый ГенераторСлучайныхЧисел();
СтрСообщения = "";
Для i=0 По N-1 Цикл
    Для j=0 По i Цикл
        A[i][j] = ГСЧ.СлучайноеЧисло (0,9);
        СтрСообщения = СтрСообщения + A[i][j] + " ";
    КонецЦикла;
    СтрСообщения = СтрСообщения +
    | " ";
КонецЦикла;
Сообщить (СтрСообщения);
```

Переменная `СтрСообщения` введена для того, чтобы вывести весь массив, а не каждый элемент в отдельности.

Заполним массив  $B$  максимальными суммами. Применим встроенную функцию `max(<Значение1>, ..., <ЗначениеN>)`. Заполненный массив выведем с помощью переменной `СтрСообщенияB`:

```

V[0][1]=A[0][0];
СтрСообщенияВ="" + V[0][1] + "
| ";
Для i=1 По N-1 Цикл
    Для j=1 По i+1 Цикл
        V[i][j]= max (A[i][j-1]+V[i-1][j],A[i][j-1]+V[i-1][j-1]);
        СтрСообщенияВ= СтрСообщенияВ+V[i][j]+" ";
    КонечЦикла;
    СтрСообщенияВ= СтрСообщенияВ+"
| "
КонечЦикла;
Сообщить (СтрСообщенияВ);

```

Найдем максимальный элемент в последней строке и сообщим его:

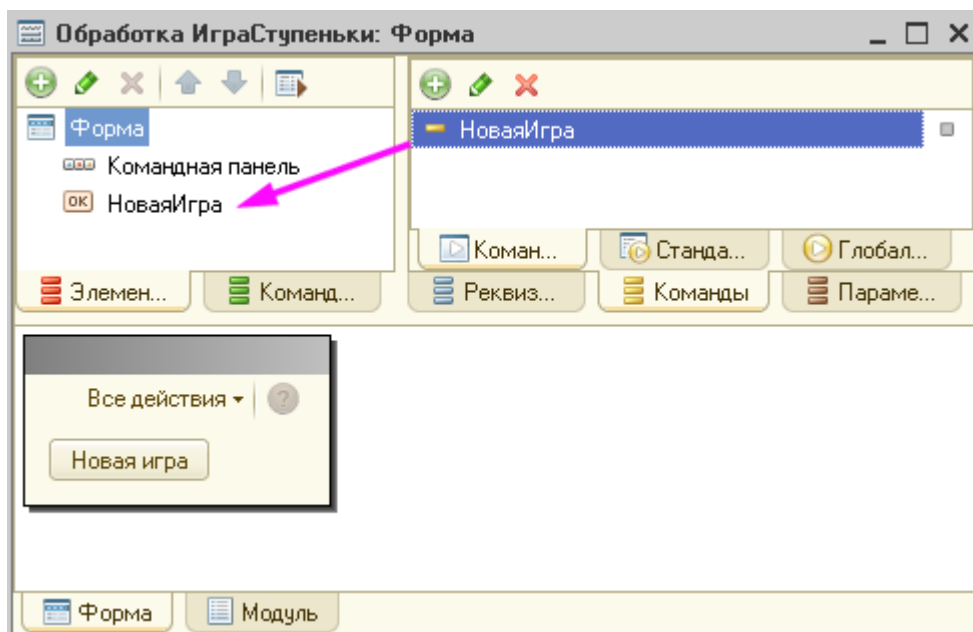
```



МахВСтроке = V[N-1][0];
Номер = 0;
Для j=1 По N-1 Цикл
    Если V[N-1][j] > МахВСтроке Тогда
        МахВСтроке = V[N-1][j];
        Номер = j;
    КонечЕсли;
КонечЦикла;
Сообщить ("Максимальное число очков "+МахВСтроке+" в "+Номер+" лузе");

```

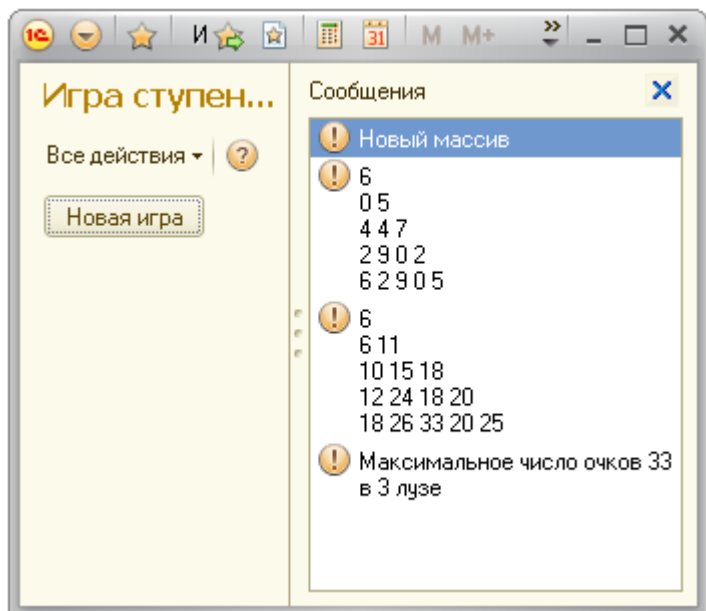
## Скачать листинг программы

7. Перейдем в окно разработки формы и перенесем кнопку команды из окна команд в окно элементов формы:



8. Перейдем в пользовательский режим:  **Начать отладку** (F5) –  **1С:Предприятие** (Ctrl+F5).

Запустим обработку *ИграСтупеньки*, введем количество уровней – 5:



Запустите несколько раз игру при  $N = 5$ .

Разработанная нами модель игры позволяет определить наилучший результат игры: количество очков и лузу, в которой должен оказаться шарик. И эта модель вполне адекватно отражает игру и соответствует поставленной цели. Если же мы захотим построить новую модель, отражающую управление шариком в ходе игры, то наша модель может стать вспомогательной.

## Настройки Конфигурации информационной системы

### Как исправить синтаксические ошибки

**Что делать, если программа зависла или требуется остановить ее из-за неправильного ввода данных?**