

# Лекция 4

Массивы, исключения, перечисления

# Массивы

```
int score1 = readInt ();
```

```
int score2 = readInt ();
```

```
int score3 = readInt ();
```

```
int score4 = readInt ();
```

```
int score5 = readInt ();
```

```
int score6 = readInt ();
```

```
int [] scores = new int [6];
```

- Массив - это логически связанный набор одного типа данных, массив int, массив double, массив string
- Количество элементов задается при объявлении массива и не может быть изменено впоследствии. Может любым выражением приводящемуся к целому числу - `int [] scores = new int[n*2 + 1];`
- Элементы массива нумеруются с нуля - `scores[0]`.

```
public static void Main (string[] args)
{
    const int SCORE_SIZE = 1000;
    int [] scores = new int[SCORE_SIZE];
    for (int i = 0; i < SCORE_SIZE; i++) {
        scores [i] = readInt ();
    }
}
```

# Двумерные массивы

```
int[,] board = new int[3, 3];
```

```
board [0, 0] = 1;
```

```
board [1, 1] = 2;
```

```
board [0, 2] = 3;
```

1	0	0
0	2	0
3	0	0

# Как упростить этот код?

```
int monthNo = 1;  
string monthName;  
if (monthNo == 1)  
    monthName = "Январь";  
if (monthNo == 2)  
    monthName = "Февраль";
```

# Lookup tables (таблицы поиска)

```
string[] monthNames = new string[12];
```

```
monthNames [0] = "Январь";
```

```
monthNames [1] = "Февраль";
```

```
monthNames [2] = "Март";
```

```
string monthName = monthNames [monthNo - 1];
```

```
string[] monthNames = new string[]  
{  
    null,  
    "Январь", "Февраль", "Март", "Апрель",  
    "Май", "Июнь", "Июль", "Август",  
    "Сентябрь", "Октябрь", "Ноябрь", "Декабрь"  
};
```

```
string monthName = monthNames [monthNo];
```



# Обработка элементов массива

```
for (int i = 1; i < monthNames.Length; i++) {  
    Console.WriteLine (monthNames [i]);  
}
```

```
foreach (string month in monthNames) {  
    if(month != null)  
        Console.WriteLine (month);  
}
```

# Свойства и методы System.Array

Length	Количество элементов
Rank	Размерность
Clear()	Присваивание элементам значений по умолчанию
IndexOf()	Индекс первого вхождения элемента в массив
Reverse()	Изменение порядка элементов
Sort()	Упорядочивание элементов

# Размер массива

```
int[] array = new int[] {1, 2, 3};  
Console.WriteLine(array.Length);
```

Количество элементов в массиве: 3

Для продолжения нажмите любую клавишу . . . 

# Ранг массива

```
int[] array1 = new int[] {1, 2, 3};  
int[,] array2 = new int[,] { { 1, 2 }, { 3, 4 } };  
Console.WriteLine("Ранг первого массива: {0}", array1.Rank);  
Console.WriteLine("Ранг второго массива: {0}", array2.Rank);
```

```
Ранг первого массива: 1  
Ранг второго массива: 2
```

# Очистка содержимого элементов

```
int[] items = new int[] { 1, 2, 3, 4, 5 };
Console.Write("Исходное состояние: ");
foreach (var i in items) {
    Console.Write("{0} ", i);
}
Console.WriteLine(" Размер: {0}", items.Length);
Array.Clear(items, 0, 3);
Console.Write("После очистки первых трех элементов: ");
foreach (var i in items) {
    Console.Write("{0} ", i);
}
Console.WriteLine(" Размер: {0}", items.Length);
```

Исходное состояние: 1 2 3 4 5    Размер: 5  
После очистки первых трех элементов: 0 0 0 4 5    Размер: 5

# Поиск элемента в массиве

```
string[] week = new string[]  
    { "", "Пн", "Вт", "Ср", "Чт", "Пт", "Сб", "Вс" };  
  
string day = "Ср";  
Console.WriteLine("{0} - {1}-й день недели", day, Array.IndexOf(week, day));
```

Ср – 3-й день недели

# Изменение порядка элементов

```
int[] items = new int[] { 1, 2, 3, 4, 5 };  
Console.Write("Исходное состояние: ");  
foreach (var i in items){  
    Console.Write("{0} ", i);  
}  
Array.Reverse(items);  
Console.Write("После вызова Array.Reverse: ");  
foreach (var i in items){  
    Console.Write("{0} ", i);  
}
```

Исходное состояние: 1 2 3 4 5

После вызова Array.Reverse: 5 4 3 2 1

# Сортировка массива

```
string[] students = new string[]  
    { "Сидоров", "Петров", "Иванов", "Афанасьев" };  
Console.Write("Исходное состояние: ");  
foreach (var s in students) {  
    Console.Write("{0}, ", s);  
}  
Console.WriteLine();  
Array.Sort(students);  
Console.Write("Отсортированный: ");  
foreach (var s in students){  
    Console.Write("{0}, ", s);  
}
```

Исходное состояние: Сидоров, Петров, Иванов, Афанасьев,  
Отсортированный: Афанасьев, Иванов, Петров, Сидоров,



# Исключения (Exceptions)

```
try {  
    int z = int.Parse (Console.ReadLine ());  
}  
catch {  
    Console.WriteLine ("Некорректный ввод");  
    return;  
}
```

```
try {  
    Console.Write("Введите возраст: ");  
    int z = int.Parse (Console.ReadLine ());  
}  
catch (Exception e) {  
    Console.WriteLine (e.Message);  
    return;  
}
```

Введите возраст: двадцать лет

Input string was not in a correct format.

Press any key to continue...

# Стандартные исключения

ArithmeticException	Ошибка в арифметических операциях
ArrayTypeMismatchException	Сохранение в массиве несовместимого типа
DivideByZeroException	Деление на ноль
FormatException	Передача аргумента неверного формата
IndexOutOfRangeException	Выход индекса массива за границы
InvalidCastException	Ошибка преобразования типа
OutOfMemoryException	Нехватка памяти
OverflowException	Переполнение при выполнении арифметических операций
StackOverflowException	Переполнение стека

```
try {  
    Console.Write("Введите номер месяца: ");  
    int monthNo = int.Parse(Console.ReadLine() );  
    Console.WriteLine(monthNames[monthNo]);  
}  
catch(FormatException){  
    Console.WriteLine ("Некорректный ввод");  
}  
catch(IndexOutOfRangeException){  
    Console.WriteLine ("Некорректный номер месяца");  
}  
catch(Exception e){  
    Console.WriteLine ("Неизвестное исключение: " + e.Message);  
}
```

```
int age = 0;
try {
    Console.Write("Введите свой возраст: ");
    age = int.Parse(Console.ReadLine() );
}
catch (FormatException) {
    Console.WriteLine ("Некорректный ввод");
    age = 20;
}
finally {
    Console.WriteLine ("Возраст = " + age);
}
```

# Свойства класса Exception

Message	Текстовое описание ошибки
Source	Имя объекта который сгенерировал ошибку
StackTrace	Стек вызовов приведших к ошибке
InnerException	Исключение послужившее причиной текущего исключения
TargetSite	Метод выбросивший исключение

```
int i = int.Parse("четыре");
```

Message	Input string was not in a correct format.
Source	mscorlib (Microsoft's Common Object Runtime Library)
StackTrace	at System.Number.StringToNumber at System.Number.ParseInt32 at System.Int32.Parse at arrays1.MainClass.Main
InnerException	
TargetSite	Void StringToNumber(...)

# Выбрасывание исключений

```
static void BuyBeer(int age) {  
    if (age < 18) {  
        throw new Exception ("Продажа запрещена!");  
    }  
    ...  
}
```



```
Console.Write("Введите свой возраст: ");
int age = int.Parse(Console.ReadLine());
try {
    BuyBeer(age);
}
catch (Exception e) {
    Console.WriteLine (e.Message);
}
finally {
    Console.WriteLine ("Возраст = " + age);
}
```

# Морской бой



# Как хранить состояние поля?

## Целочисленный массив

```
int[,] sea = new int[10, 10];  
sea[0, 1] = 1; //Уничтоженный корабль  
sea[0, 1] = 2; //Здоровый корабль  
sea[1, 1] = 3; //Пустая ячейка  
и так далее.
```

# Недостатки такого подхода

- Неинтуитивно
- Можно допустить ошибку
- Нет контроля со стороны компилятора

# Перечисления (enums)

```
enum SeaState {  
    Empty,  
    Alive,  
    Dead,  
    ...  
};  
  
SeaState[,] sea = new SeaState[10, 10];  
sea [0, 0] = SeaState.Empty;  
sea [1, 1] = SeaState.Dead;
```

# Преимущества

- Более наглядный код
- При отладке вы видите понятные значения, а не цифры
- Компилятор обнаруживает некорректные значения

```
SeaState[,] sea = new SeaState[10, 10];
```

```
sea [0, 0] = SeaState.Free;
```

```
Console.WriteLine ("Cell: " + sea [0, 0]);
```

```
//Cell: Free
```

```
sea [1, 1] = 10;
```