برنامه نویسی پیشرفته
C#

۲۴ آذر ۹۸

ملکی مجد

topics

- A program to shuffle cards
  - With arrays
  - With collections
- Action<> and Func<>
  - Use instead of defining delegate
- Lambda Expressions

Card Game:

enumeration information

```
enum Value
    {
        Two, Three, Four, Five, Six, Seven, Eight,
         Nine, Ten, Jack, Queen, King, Ace
    }
enum Suit
    {
        Clubs, Diamonds, Hearts, Spades
    }
```

Card Game: calsses

- PlayingCard
- Pack
- Hand
- Shuffle cards

Action<> and Func<>

- **Action** is a delegate (pointer) to a method, that takes zero, one or more input parameters, but does not return anything.
- **Func** is a delegate (pointer) to a method, that takes zero, one or more input parameters, and returns a value (or reference).

lambda expressions

- A *lambda expression* is an expression of any of the following two forms:
- (input-parameters) => expression
- (input-parameters) => { <sequence-of-statements> }

- Any lambda expression can be converted to a delegate type.
- If a lambda expression doesn't return a value, it can be converted to one of the Action delegate types; otherwise, it can be converted to one of the Func delegate typ

lambda expressions
simple sample

- Func<int, int, bool> testForEquality = (x, y) => x == y;

- Action<string> greet = name =>

```
{
    string greeting = $"Hello {name}!";
    Console.WriteLine(greeting);
};
greet("World");
```

example of Action

- Test4
- Convert previous example! (test2 and test3)