# Delegacje i zdarzenia

## Delegacja: idea

- Jedna wspólna konstrukcja językowa do wspierania algorytmów późnego wiązania
- Rozwiązanie *type safety*
- Wsparcie dla wywołań:
  - Single-cast
  - Multi-cast

## Delegacja: definicja

- Definicja typu:

```
public delegate
   int Comparison<in T>(T left, T right);
```

- Instancja:

```
public Comparison<T> comparator;
```

- Wywołanie:

```
int result = comparator(left, right);
```

## Predefiniowane delegacje generyczne

```
public delegate void Action();
public delegate void Action<in T>(T p);
public delegate void Action<in T1,..., in T16>
                            (T1 p1,..., T2 p16);
```

```
public delegate TR Func<out TR>();
public delegate TR Func<in T1, out TR>(T1 p);
public delegate TR Func<in T1,..., in T16,
                        out TR>(T1 p1,..., T16 p16);
```

```
public delegate bool Predicate<in T>(T p);
```

Deklaracje:

```
Action<string>     SomeAction
Func<string, bool> TestForString;
Predicate<string>  AnotherTestForString;
```

## Zdarzenia

- Zdarzenia są realizacją wzorca obserwator; obiekt będący źródłem zdarzenia powiadamia obiekty obserwujące źródło (subskrybenci)
- Wskazówki
  - Obserwator i źródło zdarzenia powinny nie być za sobą powiązani (*minimal coupling*)
  - Dostępna i nieskomplikowana procedura zapisu i rezygnacji z obserwacji zdarzenia
  - Źródło powinno obsługiwać wiele obiektów nasłuchujących (także zero)

```
public event Delegacja Zdarzenie;
```

## Zdarzenia: standardy

- Standardowa sygnatura delegacji do budowy zdarzenia:

```
void OnEventRaised(object sender, EventArgs args);
```

- Drugi argument zwykle dziedziczy po **`System.EventArgs`**,
- Jeżeli brak argmentów przekazujemy **`EventArgs.Empty`**

## Zdarzenia: standardy II

- Przy użyciu typu generycznego:

```
public event EventHandler<FileFoundArgs> FileFound;
```

- Operacje dostępne dla takiego zdarzenia:

```
EventHandler<FileFoundArgs> onFileFound =
(sender, eventArgs) =>
    Console.WriteLine(eventArgs.FoundFile);

lister.FileFound += onFileFound;

lister.FileFound -= onFileFound;
```

## Zdarzenia czy delegacje

- Zarówno zdarzenia jak i delegacje realizują późne wiązanie
- Jeżeli konstrukcja programistyczna do popranego zakończenia działania wymaga przekazania obiektu przetwarzającego: **użyj delegacji** (np. LINQ, komparator przekazany do funkcji sortującej)
- Jeżeli konstrukcja może działać poprawnie nawet bez żadnego obiektu nasłuchującego: **użyj zdarzeń**
- Jeżeli potrzebujesz kumulować operacje zwracające wartość: **użyj delegacji**