

BASIC

O/

/ I

/ \

WIGdotNET

WARSZAWSKA GRUPA .NET

PASCAL

C++

JavaScript

PHP

Assembler

JAVA

Perl Python

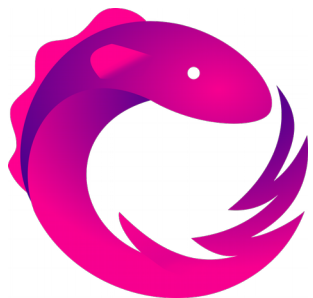
C#



Krzysztof Owsiany

GemuStudio → Alfavox

godev.gemustudio.com



Reactive Extensions - wzorzec obserwatora, czyli programowanie reaktywne.



Programowanie reaktywne

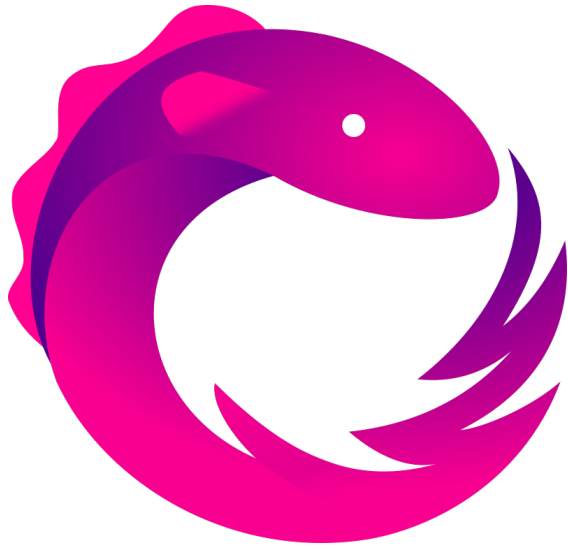
Manifest

Responsywność

Nierawność

Elastyczność

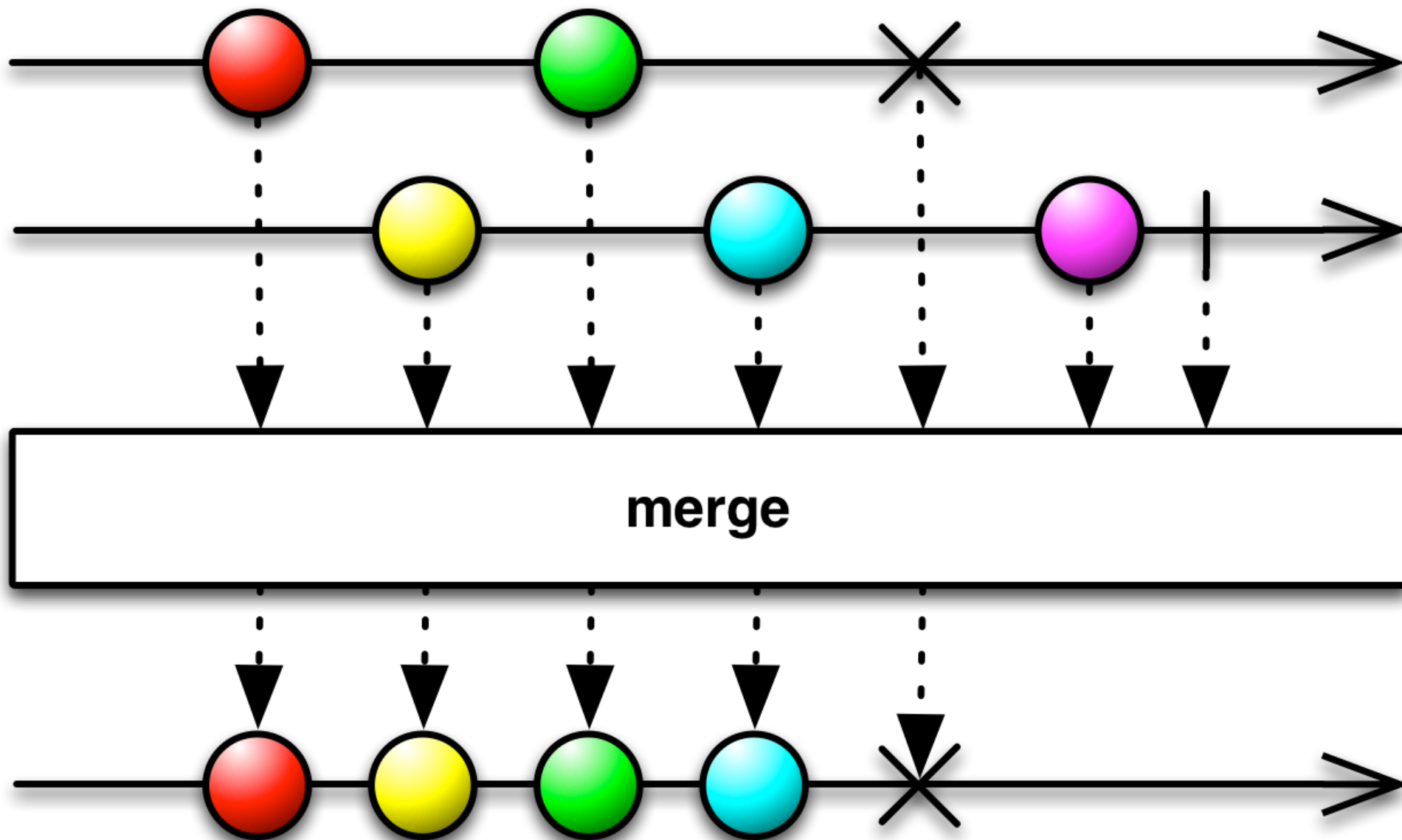
Stewardship
Komunikacji



Reactive Extensions



WIGDOTNET
WARSZAWSKA GRUPA .NET



Observable

.Kreator/Generator(lambda)

.Filter1(lambda)

.Modyfikator1(lambda)

.Selektor1(lambda)

.Modyfikator2(lambda)

.Filter2(lambda)

.Selektor2(lambda)

.Subscribe(
 OnNext,
 OnError,
 OnCompleted
);

```
var stream =  
    Observable  
        .Interval(TimeSpan.FromSeconds(1));  
  
var subscriber = stream  
    .Subscribe(  
        tick => Debug.WriteLine(tick)  
    );
```

```
var stream =  
    Observable.Interval (  
        TimeSpan.FromSeconds(1),  
        NewThreadScheduler.Default  
    )  
    .Timestamp();
```

```
var subscribent = stream  
    .ObserveOn(Dispatcher.CurrentDispatcher)  
    .Subscribe(  
        tick => Label1.Content = tick  
    );
```

```
var stream =  
    Observable  
        .FromEventPattern<TextChangedEventHandler,  
TextChangedEventArgs>(  
            @event => TextBox1.TextChanged += @event,  
            @event => TextBox1.TextChanged -= @event  
        )  
        .Select(x => x.Sender)  
        .OfType<TextBox>()  
        .Select(  
            textBox => textBox.Text  
        );
```

Observable

```
.FromEventPattern<RoutedEventHandler,RoutedEventArgs>(  
    @event => ExitButton.Click += @event,  
    @event => ExitButton.Click -= @event  
)  
.Subscribe(  
    eventArgs => Action()  
);
```

```
var stream =  
    Observable  
        .FromEventPattern<TextChangedEventHandler,  
TextChangedEventArgs>(   
            @event => TextBox1.TextChanged += @event,  
            @event => TextBox1.TextChanged -= @event  
        )  
        .Select(x => x.Sender)  
        .OfType<TextBox>()  
        .Select(textBox => textBox.Text);
```

```
var subscribent = stream  
    .Where(text =>  
        text.ToLower().Equals("fibonacci")  
    ).Subscribe(  
        text => Action()  
    );
```



```
public class EventObservable : IObservable<IEvent>
{
    private IList<IObserver<IEvent>> _subscribers;

    public EventObservable(){
        _subscribers = new List<IObserver<IEvent>>();
    }

    public void Push(IEvent eventType){
        foreach (var subscriber in _subscribers){
            subscriber.OnNext(eventType);
        }
    }

    public IDisposable Subscribe(IObserver<IEvent> observer){
        if (!_subscribers.Contains(observer)){
            _subscribers.Add(observer);
        }

        return new Unsubscribe<IEvent>(observer, _subscribers);
    }
}
```

```
public class Unsubscribe<T> : IDisposable
{
    private readonly IObserver<T> _observer;
    private readonly IList<IObserver<T>> _subscribers;

    public Unsubscribe(
        IObserver<T> observer,
        IList<IObserver<T>> subscribers)
    {
        _observer = observer;
        _subscribers = subscribers;
    }

    public void Dispose()
    {
        _subscribers.Remove(_observer);
    }
}
```

```
public class ExitObserver : IObserver<ApplicationExitedEvent>
{
    public void OnNext(
        ApplicationExitedEvent applicationExitedEvent)
    {
        Debug.WriteLine("Observer: ExitObserver");

        Environment.Exit(applicationExitedEvent.ExitCode);
    }

    public void OnCompleted() { }

    public void OnError(Exception error) { }
}
```

```
    _name = name;  
    Initialize(name);  
    _timerObservable =  
        Observable.Timer(_dueTime, _period, scheduler);  
    Subscribe();  
}  
  
private void Initialize(String  
    {  
        _dueTime = TimeSpan.  
        _period = TimeSpan.  
        _name = name;  
    }
```

CQRS/ES

GUI – obsługa kontrolek

Zdarzenia w domenie

Klient-Serwer

Nasłuch urządzeń peryferyjnych

Model Aktorów

reactivex.io

reactivemanifesto.org

rxmarbles.com

introtorx.com

rxwiki.wikidot.com/101samples

godev.gemustudio.com

github.com/Reactive-Extensions/Rx.NET

github.com/godevblog/30DayChallenge

github.com/krzysztofowsiany/rxexamples

Krzysztof Owsiany

Twitter: **@k_owsiany**

Blog: **mrdev.pl**

Snapchat: **gocom7**

