

# WALIDACJA W STYLU FUNKCYJNYM

# Niech dane będą dane

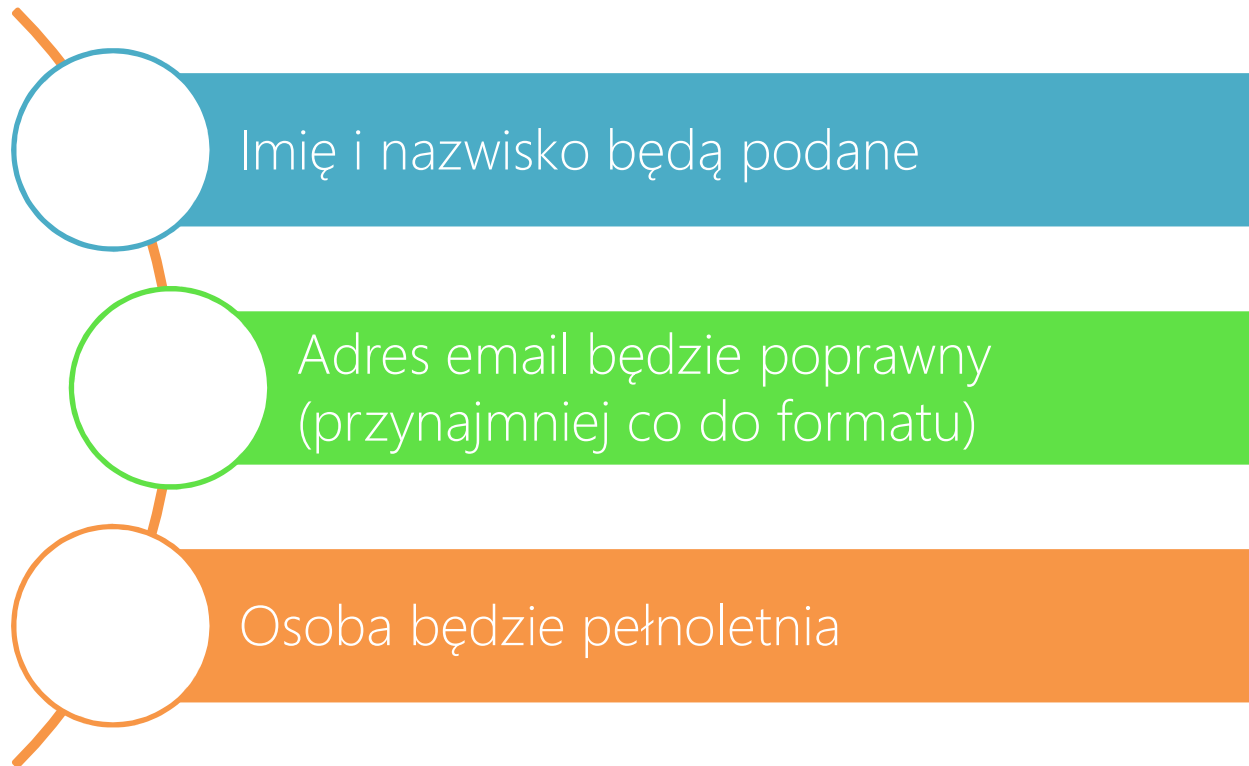
---

```
type Osoba = {  
  id:int  
  imie:string  
  nazwisko:string  
  email:string  
  wiek:int  
}
```

```
let stworzOsobe id imie nazwisko email wiek =  
  {id= id;  
   imie = imie;  
   nazwisko = nazwisko;  
   email = email;  
   wiek=wiek}
```

# Chcemy mieć pewność, że:

---



## Postać walidatora

---

```
let pelnoletnia wiek =  
    if wiek >= 18 then Ok(wiek)  
    else Error("Osoba musi być pełnoletnia")
```

'TypWartosci -> Result<'TypWartosci, string>

# Walidatory

---

```
let stringNiePusty (str:string) =  
    if not (String.IsNullOrEmpty(str)) then Ok(str)  
    else Error("String nie może być pusty")  
  
let pelnoletnia wiek =  
    if wiek >= 18 then Ok(wiek)  
    else Error("Osoba musi być pełnoletnia")  
  
let emailPoprawny email =  
    let wzorzec = @"^([\w\.\-]+)@([\w\-]+)((\.(\w){2,3})+)$"  
    if Regex.IsMatch(email, wzorzec) then Ok(email)  
    else Error("To musi być poprawny adres email")
```

## Połączenie za pomocą bind

---

```
let walidujOsobe id imie nazwisko email wiek =  
  stringNiePusty imie  
  |> Result.bind (fun _-> stringNiePusty nazwisko)  
  |> Result.bind (fun _-> emailPoprawny email)  
  |> Result.bind (fun _-> pelnoletni wiek)  
  |> Result.bind (fun _-> Ok(stworzOsobe id  
                                     imie  
                                     nazwisko  
                                     email  
                                     wiek)  
                                     )
```

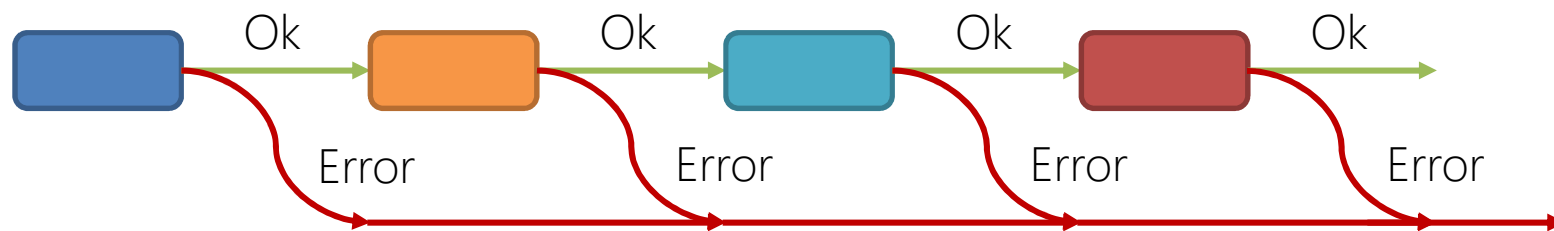
## Połączenie za pomocą bind

---

```
let walidujOsobe id imie nazwisko email wiek =  
  stringNiePusty imie  
  >>= (fun _ -> stringNiePusty nazwisko)  
  >>= (fun _ -> emailPoprawny email)  
  >>= (fun _ -> pelnoletni wiek)  
  >>= (fun _ -> Ok(stworzOsobe id imie nazwisko email wiek))
```

## Problem z tym sposobem walidacji

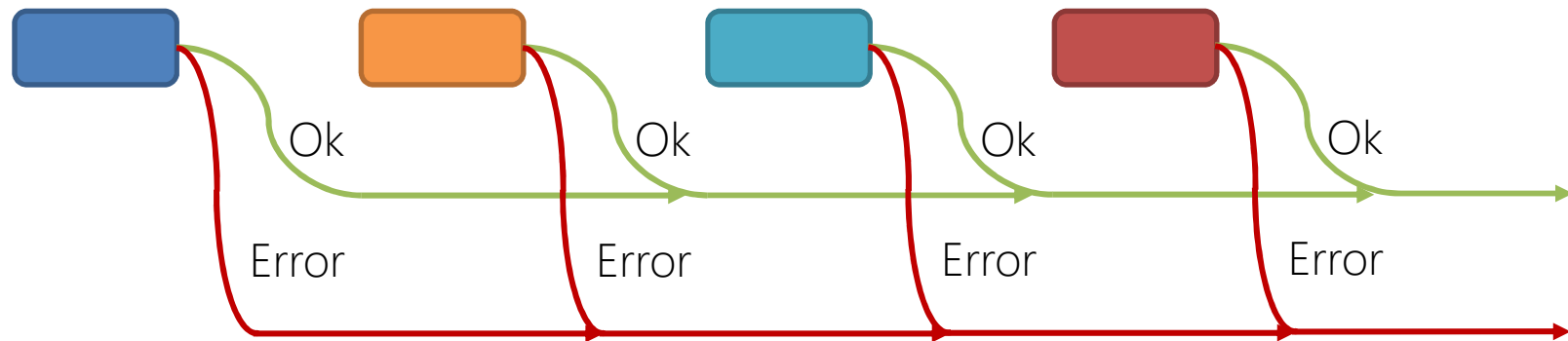
---





## Problem z tym sposobem walidacji

---



```
stworzOsobe id (stringNiePusty imie)
              (stringNiePusty nazwisko)
              (emailPoprawny email)
              (pelnoletnia wiek)
```

# Aplikatywy

---

f: int->cokolwiek

opcje: int option

mapuj:  $M \langle 'T \rangle \rightarrow ('T \rightarrow 'R) \rightarrow M \langle 'R \rangle$

```
let mapuj mapa opcja =  
  match opcja with  
  | Some x -> Some (mapa x)  
  | None -> None
```

# Aplikatywy

```
let dodaj a b = a+b
```

mapuj dodaj (Some 1) ➡

```
match (Some 1) with
| Some x -> Some (dodaj x)
| None -> None
```

# Aplikatywy

---

```
let aplikuj param funkcja =  
  match funkcja, param with  
  | Some fn, Some v -> Some(fn v)  
  | None, Some _ -> None  
  | Some _, None -> None  
  | None, None -> None
```

aplikuj:  $M \langle ' T \rangle \rightarrow M \langle ' T \rightarrow ' R \rangle \rightarrow M \langle ' R \rangle$

## Aplikatywy

---

aplikuj:  $M \langle ' T \rangle \rightarrow M \langle ' T \rightarrow ' R \rangle \rightarrow M \langle ' R \rangle$

```
Some(dodaj)  
|> aplikuj (Some 1)  
|> aplikuj (Some 2)
```

```
mapuj dodaj (Some 1)  
|> aplikuj (Some 2)
```

## Aplikatywy

---

aplikuj:  $M \langle ' T \rangle \rightarrow M \langle ' T \rightarrow ' R \rangle \rightarrow M \langle ' R \rangle$

```
let (<*>) funkcja wartosc = mapuj funkcja wartosc  
let (<**>) funkcja wartosc = aplikuj wartosc funkcja
```

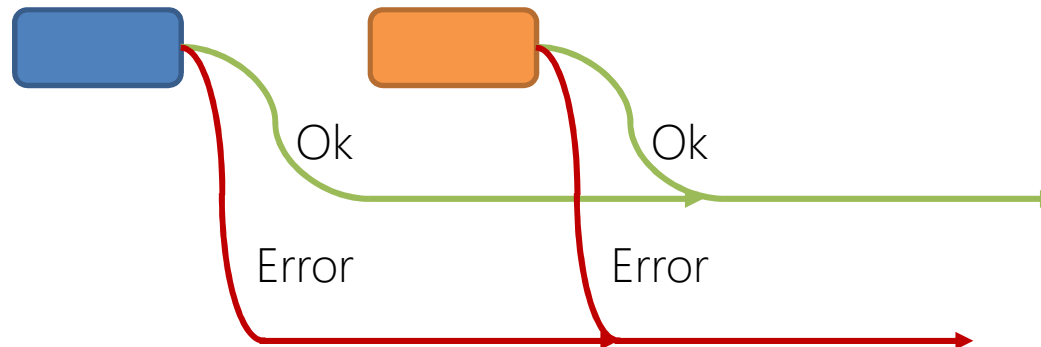
```
Some(dodaj) <**> (Some 1) <**> (Some 2)
```

```
dodaj <*> (Some 1) <**> (Some 2)
```

# Aplikatywy - Result

---

```
let aplikuj parametr funkcja =  
  match funkcja, parametr with  
  | Ok fn, Ok v -> Ok(fn v)  
  | Error e, Ok _ -> Error e  
  | Ok _, Error e -> Error [e]  
  | Error e1, Error e2 -> Error (e2::e1)
```



## Aplikatywy - Result

---

```
let aplikuj parametr funkcja =  
  match funkcja, parametr with  
  | Ok fn, Ok v -> Ok(fn v)  
  | Error e, Ok _ -> Error e  
  | Ok _, Error e -> Error [e]  
  | Error e1, Error e2 -> Error (e2::e1)
```

```
let (<**>) funkcja wartosc = aplikuj wartosc funkcja
```



## Walidacja za pomocą apply

---

```
let walidujOsobe id imie nazwisko email wiek =  
  Ok (stworzOsobe id)  
    |> aplikuj (stringNiePusty imie)  
    |> aplikuj (stringNiePusty nazwisko)  
    |> aplikuj (emailPoprawny email)  
    |> aplikuj (pelnoletni wiek)
```

```
let walidujOsobe id imie nazwisko email wiek =  
  Ok(stworzOsobe id)  
    <**> stringNiePusty imie  
    <**> stringNiePusty nazwisko  
    <**> emailPoprawny email  
    <**> pelnoletni wiek
```