



Kontynuacje i ich zastosowanie do WWW

Jan Rychter

<http://jan.rychter.com/blog/>



Kontynuacje: co to jest?

- Kontynuacja to stan programu od pewnej chwili do końca jego wykonania
- Podobne do wyjątków (catch/throw)
- Różnica: kontynuacje można zachowywać i wołać z dowolnego miejsca programu



Prawie jak kontynuacje

(rób-coś)

(wypisz (catch 'k (jakaś-funkcja))

(rób-coś-jeszcze)



Prawie jak kontynuacje

(rób-coś)

(wypisz (catch 'k (jakaś-funkcja)))

(rób-coś-jeszcze)

(define (jakaś-funkcja)

(throw 'k 42))



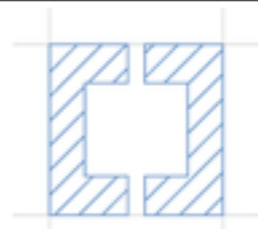
Magiczny operator call/cc

(rób-coś)

(wypisz (call/cc moja-funkcja))

(rób-coś-jeszcze)

- call/cc = call-with-current-continuation

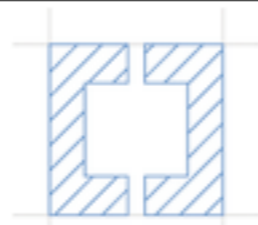


Tech Aula

(rób-coś)

(wypisz | (call/cc moja-funkcja))

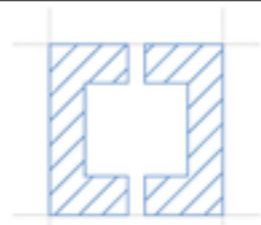
(rób-coś-jeszcze)



Tech Aula

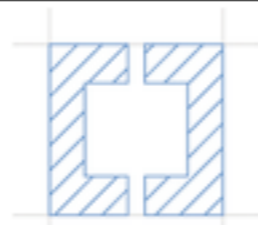
(rób-coś)
(wypisz | (call/cc moja-funkcja))
(rób-coś-jeszcze)

kontynuacja

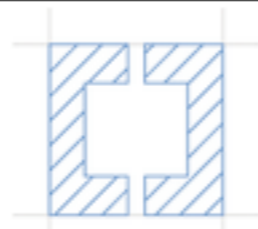


Tech Aula





Tech Aula

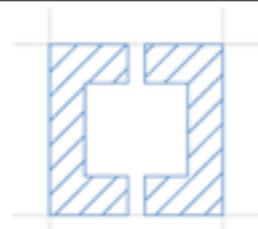


Tech Aula

(rób-coś)

(wypisz (call/cc moja-funkcja))

(rób-coś-jeszcze)



Tech Aula

```
(define (k a)
  (wypisz a)
  (rób-coś-jeszcze))
```



```
(rób-coś)
(wypisz | (call/cc moja-funkcja))
(rób-coś-jeszcze)
```

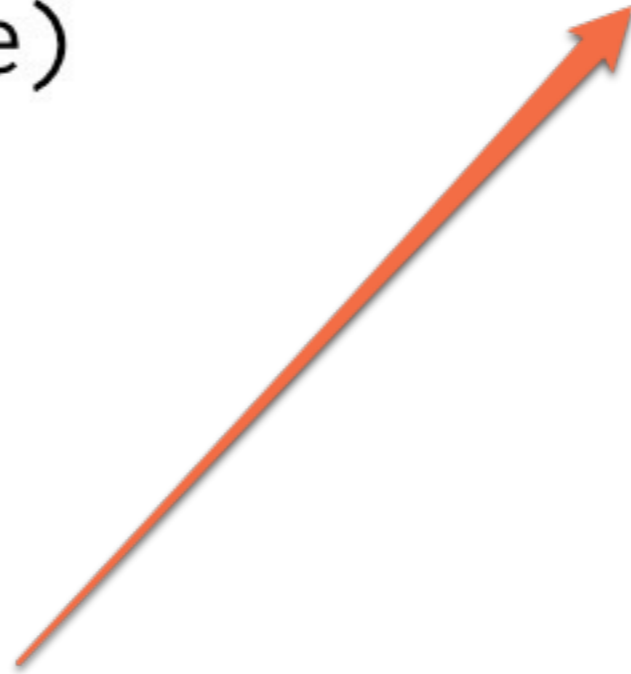


```
(define (k a)
  (wypisz a)
  (rób-coś-jeszcze))
```

```
(rób-coś)  
(wypisz | (call/cc moja-funkcja))  
(rób-coś-jeszcze)
```



```
(define (k a)  
  (wypisz a)  
  (rób-coś-jeszcze))
```



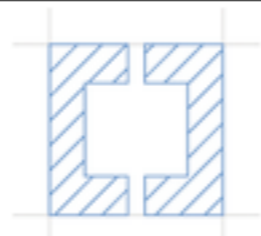


Co zrobić z kontynuacją?

- Zawołać:

```
(define (moja-funkcja k)  
  (k 42))
```
- Zachować na później:

```
(define (moja-funkcja k)  
  (zachowaj-w-danych-sesji k)  
  (wyświetl-stronę-www)  
  (czekaj-na-żądanie-http))
```



Tech Aula



W aplikacjach WWW

- Wracamy do sensownego pisania programów:
(rób-coś)
(if (and (zgoda-na-warunki1)
 (zgoda-na-warunki2))
 (wyświetl-tajną-stronę)
 (podziękuj-grzecznie))
- Możliwe dzięki zachowywaniu kontynuacji
- Przy kolejnym żądaniu HTTP wołamy kontynuację



Kto używa kontynuacji?

- Seaside (Smalltalk)
- PLT (Scheme)
- UCW (Common Lisp)
- Weblocks (Common Lisp)
- Apache Cocoon (Java+Javascript)



Prawdziwy kod: Weblocks

```
(wyświetl-dane x)
(link/cc "Skasuj...")
  (if (do-confirmation "Czy na pewno?")
      (skasuj x))
```

Prawdziwy kod: Cocoon

```
function calculator()  
{  
    var a, b, operator;  
  
    cocoon.sendPageAndWait("getA.html");  
    a = cocoon.request.get("a");  
  
    cocoon.sendPageAndWait("getB.html");  
    b = cocoon.request.get("b");  
    [...]
```



Prawdziwy kod: Seaside

```
go
```

```
[ self chooseCheese.  
  self confirmCheese ] whileFalse.  
self informCheese
```

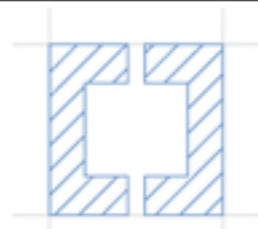
```
confirmCheese
```

```
  ^ self confirm: 'Is ', cheese , 'your  
favorite Cheese?'
```



Gdzie dalej?

- Zacząć najlepiej od Seaside
- Dla początkujących lispowców: PLT Scheme
- Dla zaawansowanych lispowców, co symboli się nie boją: Weblocks
- Dla twardych: Apache Cocoon



Tech Aula

?