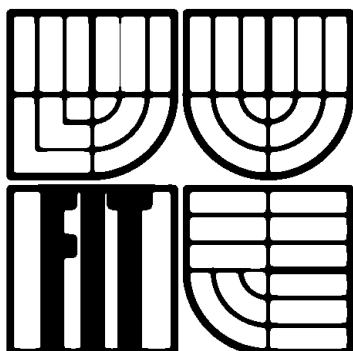
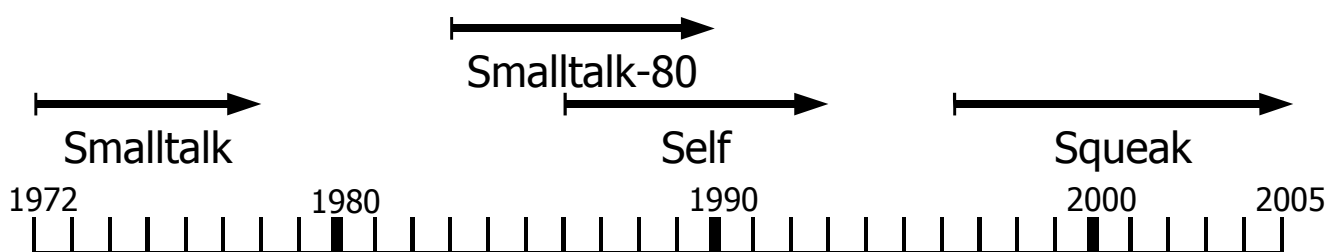


Podpora beztřídního programování ve Squeak Smalltalku



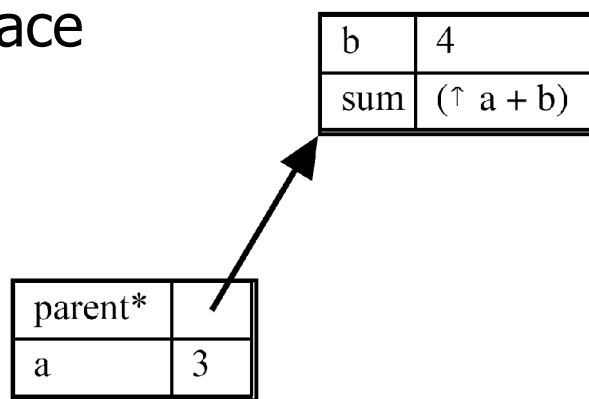
řešitel: Pavel Křivánek
vedoucí: Janoušek Vladimír, Ing., Ph.D.

nenavazuje na žádný RP ani SP



Self

- beztřídní objektová orientace
- objekt = množina slotů
- delegace
- dynamická dědičnost
- třída ~ rys + prototyp



sum = 7

Self & Squeak

- MVC - Morphic (GUI)
- eToys
 - vizuální programování
 - pro každý scriptovaný morph nová třída!

„Simplicity can be marvellously powerful“ - Rahul Jindal

Přínos Selfu oproti Smalltalku

- jednodušší, promyšlenější, mocnější
- dynamická a násobná dědičnost, mixins
- jmenné prostory, modularizace image

Negativa

- projekt na pokraji zájmu společnosti Sun
- pomalá VM pro Linux, žádná pro Windows
- malá komunita, multimédia, absence Seaside...

Squeak a prototypy

- několik implementací
- neefektivní, problematická syntaxe...
- pro větší projekty nepoužitelné

Proč ne Smalltalk

- třídní jazyk
- pseudoproměnné, literály...

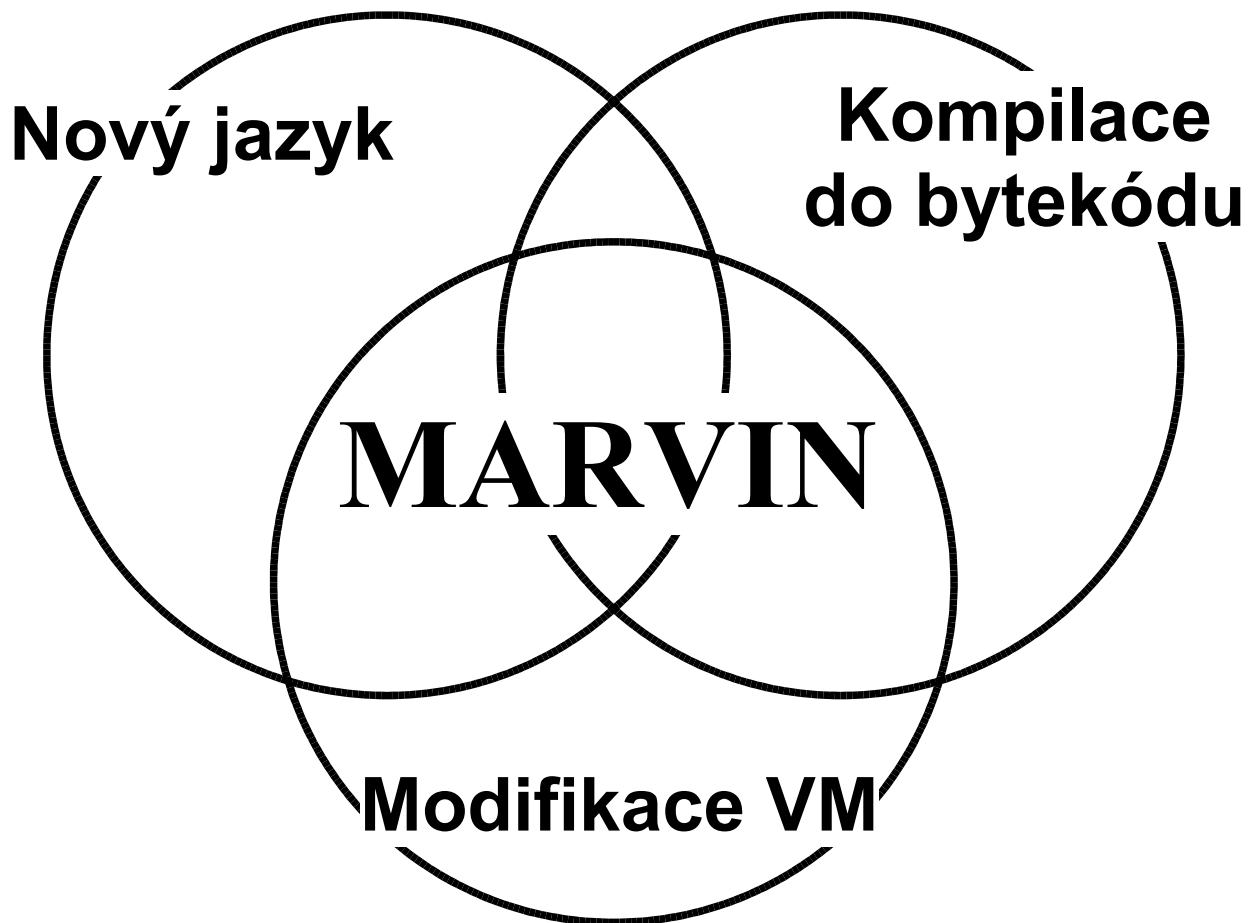
Proč ne Self, Slate apod.

- problematická integrace se Squeakem
- neefektivní implementace

Nový komplexní přístup

kombinace Smalltalku a Selfu
literály pro zápis objektů a slotů
smalltalkovské konvence pro zprávy
smalltalkovské literály

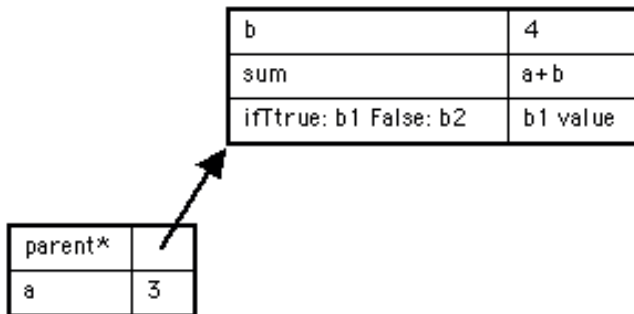
přímo bytekód Squeaku
ekvivalentní se Smalltalkem
smalltalkovské bloky
lokální sloty jako proměnné
úplná integrace se Squeakem



podpora prototypů
podpora delegace
podpora přeposílání zpráv
samostatně použitelné

Marvin vs. Self

- plně srovnatelné vyjadřovací schopnosti
- lepší podpora bloků, čísel, češtiny, jmenných prostorů...
- plná provázanost se Squeakem



Self

```
( |
  parent* = ( |
    b = 4.
    sum = (a+b).
    ifTrue: b1 False: b2 = ( b1 value ) | ).
  a = 3.
| )
```

Marvin

```
( |
  parent* = [( |
    b = [4].
    sum = (^ a+b).
    ifTrue: b1 iffFalse: b2 = ( ^ b1 value ) | )].
  a = [3].
| )
```

Smalltalk

```
Object subclass: #Parent1
  instanceVariableNames: 'b'
  classVariableNames: ''
  poolDictionaries: ''
  category: 'Demo'
```

```
!Parent1 methodsFor: 'as yet unclassified' stamp: 'pk 6/16/2005 16:16!'
b
```

```
^ b! !
```

```
!Parent1 methodsFor: 'as yet unclassified' stamp: 'pk 6/16/2005 16:17!'
ifTrue: b1 iffFalse: b2
```

```
^ b1 value! !
```

```
!Parent1 methodsFor: 'as yet unclassified' stamp: 'pk 6/16/2005 16:19!'
initialize
```

```
b := 4! !
```

```
!Parent1 methodsFor: 'as yet unclassified' stamp: 'pk 6/16/2005 16:17!'
sum
```

```
^ self a + self b! !
```

```
Parent1 subclass: #Object1
  instanceVariableNames: 'a'
  classVariableNames: ''
  poolDictionaries: ''
  category: 'Demo'
```

```
!Object1 commentStamp: 'pk 6/16/2005 16:20' prior: 0!
Object1 new!
```

```
!Object1 methodsFor: 'as yet unclassified' stamp: 'pk 6/16/2005 16:18!'
a
```

```
^ a! !
```

```
!Object1 methodsFor: 'as yet unclassified' stamp: 'pk 6/16/2005 16:19!'
initialize
```

```
a := 3! !
```

LZE VYUŽÍVAT SQUEAKOVSKÉ OBJEKTY A TŘÍDY a naopak

Další vývoj

- outliner, debugger a další vývojové nástroje
- optimalizace
- prototypizace image
- 3D víceuživatelské pracovní prostředí