

Programovací jazyk

Cobra

Matěj Sedláček, Václav Jiříček

18.12. 2012

Obsah

- **Co je Cobra?**
- **Proč Cobra vznikl?**
- **Co Cobra umí?**
- **Co Cobra ještě neumí?**
- **Závěr?**

Co je Cobra?

- obecně zaměřený OOP jazyk
- Autorem je Chuck Norris, z LA

Co je Cobra?

- obecně zaměřený OOP jazyk
- Autorem je Chuck Esterbrook, z LA

Co je Cobra?

- obecně zaměřený OOP jazyk
- Autorem je Chuck Esterbrook, z LA
- Počátky na podzim 2006

Co je Cobra?

- obecně zaměřený OOP jazyk
- Autorem je Chuck Esterbrook, z LA
- Počátky na podzim 2006
- **Cobra** = **C**oding for **b**eginners and **r**eal **a**mateurs ?? Asi ne.

Proč Cobra vznikl?

- Snaha dát dohromady výhody ostatních programovacích jazyků.

Proč Cobra vznikl?

- Snaha dát dohromady výhody ostatních programovacích jazyků.
- Ovlivněn převážně jazyky: Python, C#, Eiffel, Objective-C.

Snaha o ...

- Jednoduchou a přehlednou syntaxy (Python, Ruby)

Snaha o ...

- Jednoduchou a přehlednou syntaxi (Python, Ruby)
- Run-time performance (C#, C++)

Snaha o ...

- Jednoduchou a přehlednou syntaxy (Python, Ruby)
- Run-time performance (C#, C++)
- Static and Dynamic typing (Objective-C, VB)

Snaha o ...

- Jednoduchou a přehlednou syntaxy (Python, Ruby)
- Run-time performance (C#, C++)
- Static and Dynamic typing (Objective-C, VB)
- Contracts (Eiffel, Spec#)

Snaha o ...

- Jednoduchou a přehlednou syntaxy (Python, Ruby)
- Run-time performance (C#, C++)
- Static and Dynamic typing (Objective-C, VB)
- Contracts (Eiffel, Spec#)
- Nil tracking (Spec#)

Snaha o ...

- Jednoduchou a přehlednou syntaxy (Python, Ruby)
- Run-time performance (C#, C++)
- Static and Dynamic typing (Objective-C, VB)
- Contracts (Eiffel, Spec#)
- Nil tracking (Spec#)
- Unit tests

Snaha o ...

- Jednoduchou a přehlednou syntaxy (Python, Ruby)
- Run-time performance (C#, C++)
- Static and Dynamic typing (Objective-C, VB)
- Contracts (Eiffel, Spec#)
- Nil tracking (Spec#)
- Unit tests
- Doc strings (Python)

Co Cobra umí?

- Jednoduchou a přehlednou syntaxi (Cobra, Python, Ruby)
- Run-time performance (Cobra, C#, C++)
- Static and Dynamic typing (Cobra, VB)
- Contracts (Cobra, Eiffel, Spec#)
- Nil tracking (Cobra, Spec#)
- Unit tests
- Doc strings (Cobra, Python)

Pythonu podobná syntaxe

```
print "Hello, World!"
```

```
class Hello
```

```
    def main
```

```
        print 'Hello, world.'
```

Statické a dynamické typy, statická a dynamická vazba

```
def add(a as int, b as int) as int
```

```
def add(a, b) as dynamic
```

Programmers should choose
not language designers

Doc strings – dokumentace

```
class Utils
  shared
    def countChars(s as String, c as char) as int
      """
      Returns the number of instances of c in s.
      """
      count = 0
      for ch in s
        if c == ch
          count += 1
      return count
```

Doc strings – dokumentace

- Dokumentace metody přímo pod její deklarací.
- Zavolat pomocí `Utils.countChars._doc`.
- Jinak lze dokumentovat pomocí `#...` nebo `/#...#/`.

Unit tests – Unit testy

- Kontroluje zda třída, vlastnost či metoda dělají co mají.
- Přimo u kódu, který kontroluje.
- Dokumentuje kód.

Unit tests – Unit testy

```
class Utils
  shared
  def countChars(s as String, c as char) as int
    """
    Returns the number of instances of c in s.
    """
    test
      assert Utils.countChars('', c'x') == 0
      assert Utils.countChars('x', c'x') == 1
      assert Utils.countChars('X', c'x') == 0 # case sensitive
      assert Utils.countChars('!! ', c'!') == 2
    body
      count = 0
      for ch in s
        if c == ch
          count += 1
      return count
```

Contracts – Kontrakty

- Další možnost kontroly správného fungování metody.
- Popisuje vstupní a výstupní podmínky pro danou metodu.
- Dokumentuje kód.

Contracts – Kontrakty

```
class Customer

  var _contacts as List<of Contact>

  get contacts from var

  def addContact(contact as Contact)
    require
      contact not in .contacts
      contact.name
      contact.customer is nil
    ensure
      contact.customer == this
      .contacts.count == old .contacts.count + 1
    body
      contact.customer = this
      _contacts.add(contact)
```

Nil tracking – sledování nilu

- Předávání nilu, tam kam to není žádoucí generuje těžko dohledatelné chyby.
- Rozlišení na nil podporující a nepodporující typy pomocí otazníku (?).
- Zachycení nilu již při kompilaci.

Nil tracking – sledování nilu

```
def countChars(s as String, c as char) as int
```

```
Utils.countChars(nil, c'x')
```

```
class Foo
```

```
  def bar(s as String?)
```

```
    if s is not nil
```

```
      print Utils.countChars(s, c'x')
```

Výhody ke skriptování

- jednoduchá syntaxe

Výhody ke skriptování

- jednoduchá syntaxe
- kompilace a spuštění programu v jednom příkazu: `> cobra priklad.cobra`

Výhody ke skriptování

- jednoduchá syntaxe
- kompilace a spuštění programu v jednom příkazu: `> cobra priklad.cobra`
- shebang řádka (`#!`) u Unixových systému
`#!/usr/bin/cobra`

Výhody ke skriptování

- jednoduchá syntaxe
- kompilace a spuštění programu v jednom příkazu: `> cobra priklad.cobra`
- shebang řádka (`#!`) u Unixových systému
`#!/usr/bin/cobra`
- dynamická vazba

Co ještě Cobra umí?

- Přesnější matematika - podpora datových typů *float* (64/32-bit) i *decimal* (128-bit).

Co ještě Cobra umí?

- Přesnější matematika - podpora datových typů *float* (64/32-bit) i *decimal* (128-bit).
- mixins ~ abstraktní třídy, s možností vícenásobného dědění.

Co ještě Cobra umí?

- Přesnější matematika - podpora datových typů *float* (64/32-bit) i *decimal* (128-bit).
- mixins ~ abstraktní třídy, s možností vícenásobného dědění.
- Vytváří binární/bytové kódy zpracovatelné ostatními jazyky(C#, VB, Java,...).

Co zatím chybí?

- Výstup (ve formě bytových-kódů) pro JVM (Java Virtual Machine).

Co zatím chybí?

- Výstup (ve formě bytových-kódů) pro JVM (Java Virtual Machine).
- Zatím jen základní vývojové prostředí Visual Cobra Express.

Závěrem

- Běží na Microsoft .NET framework 2.0+ na Windows, nebo na Novell Mono 2.6+ na Linux, MacOS X.
- Open-sourceový programovací jazyk.
- Stavěný na středně velké a velké projekty.
- 17.11. 2006 vyšla Cobra 0.2 – kompilátor Cobry napsaný v Cobře.
- Nyní verze 0.9.3.

Děkuji za pozornost.

cobra-language.com