

КОНСТРУКЦИИ ЗА РАБОТА С МНОЖЕСТВА ОТ ЕКЗЕМПЛЯРИ

COOL предоставя средства (вградени функции) за определяне и извършване на действия с множества от екземпляри на потребителски класове, които удовлетворяват определени ограничения, дефинирани от потребителя. Ограниченията се задават под формата на булеви изрази.

Примерни дефиниции

```
> (defclass PERSON (is-a USER) (role abstract)
    (slot sex (access read-only)
            (storage shared))
    (slot age (type NUMBER)
            (visibility public)))

> (defmessage-handler PERSON put-age (?value)
    (dynamic-put age ?value))

> (defclass FEMALE (is-a PERSON) (role abstract)
    (slot sex (source composite)
            (default female)))
```

```
> (defclass MALE (is-a PERSON) (role abstract)
    (slot sex (source composite)
              (default male)))
> (defclass WOMAN (is-a FEMALE) (role concrete)
    (slot age (source composite) (default 25)
              (range 18.0 100.0)))
> (defclass MAN (is-a MALE) (role concrete)
    (slot age (source composite) (default 25)
              (range 18.0 100.0)))
```

```
> (definstances PEOPLE
    (Man-1 of MAN (age 18))
    (Man-2 of MAN (age 60))
    (Woman-1 of WOMAN (age 18))
    (Woman-2 of WOMAN (age 60))
    (Woman-3 of WOMAN))
> (reset)
```

- Проверка дали съществува множество от екземпляри, които удовлетворяват дадено ограничение

Синтаксис:

**(any-instancep <instance-set-template>
<query>)**

Проверява дали съществува множество от екземпляри, които съответстват на шаблона **<instance-set-template>** и удовлетворяват ограничението **<query>**. Връща стойност TRUE или FALSE в зависимост от успеха на проверката.

- **Определяне на множество от екземпляри, които удовлетворяват дадено ограничение**

Синтаксис:

**(find-instance <instance-set-template>
<query>)**

Връща като резултат първото намерено множество от екземпляри, които съответстват на шаблона **<instance-set-template>** и удовлетворяват ограничението **<query>**.

Пример. Намиране на първата двойка от мъж и жена на една и съща възраст.

```
> (find-instance ((?m MAN) (?w WOMAN))  
    (= ?m:age ?w:age))  
( [Man-1] [Woman-1] )
```

- **Определяне на всички множества от екземпляри, които удовлетворяват дадено ограничение**

Синтаксис:

```
(find-all-instances  
  <instance-set-template> <query>)
```

Връща като резултат множествена (multifield) стойност, съдържаща елементите на всички намерени множества от екземпляри, които съответстват на шаблона **<instance-set-template>** и удовлетворяват ограничението **<query>**.

Пример. Намиране на всички двойки от мъж и жена на една и съща възраст.

```
> (find-all-instances  
   ((?m MAN) (?w WOMAN))  
   (= ?m:age ?w:age))  
([Man-1] [Woman-1] [Man-2] [Woman-2])
```

- **Изпълнение на определено действие върху множество от екземпляри, които удовлетворяват дадено ограничение**

Синтаксис:

**(do-for-instance <instance-set-template>
<query> <action>)**

Изпълнява действието **<action>** върху първото намерено множество от екземпляри, които съответстват на шаблона **<instance-set-template>** и удовлетворяват ограничението **<query>**.

Пример. Отпечатване на първата двойка от различни хора на една и съща възраст.

```
> (do-for-instance  
    ((?p1 PERSON) (?p2 PERSON))  
    (and (= ?p1:age ?p2:age)  
          (neq ?p1 ?p2))  
    (printout t ?p1 " " ?p2 crlf))  
[Man-1] [Woman-1]
```

- **Изпълнение на определено действие върху всички множества от екземпляри, които удовлетворяват дадено ограничение**

Синтаксис:

```
(do-for-all-instances  
  <instance-set-template>  
  <query> <action>)
```

Изпълнява действието **<action>** върху всички намерени множества от екземпляри, които съответстват на шаблона **<instance-set-template>** и удовлетворяват ограничението **<query>**.

Пример. Отпечатване на всички двойки от хора на една и съща възраст. Обръщението към **str-compare** е направено с цел в резултата да не се включват двойки, които се различават само по реда на елементите си.

```
> (do-for-all-instances
    ((?p1 PERSON) (?p2 PERSON))
    (and (= ?p1:age ?p2:age)
          (> (str-compare ?p1 ?p2) 0))
    (printout t ?p1 " " ?p2 crlf))
[Woman-1] [Man-1]
[Woman-2] [Man-2]
```