

没有RailsJava程序员如何使用Ruby[2] PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/245/2021_2022__E6_B2_A1_E6_9C_89Rail_c104_245777.htm 五、快速移动对象

在Ruby中，创建对象实例的方法与Java不同。不是使用Java中的"new ObjectInstance()"语法，而是，Ruby支持在一个对象上调用一个新方法，实际上是在内部调用了initialize方法。在列表4中，你可以看到我是怎样用Ruby创建一个Word实例及一些相应的Definition的：

```
列表4.在Ruby中创建一个新对象实例
require "dictionary"
happy_wrd = Dictionary::Word.new("ebullient", "adjective")
defin_one = Dictionary::Definition.new("Overflowing with enthusiasm")
defin_two = Dictionary::Definition.new("Boiling up or
```

```
over")
happy_wrd.add_definition(defin_one)
happy_wrd.add_definition(defin_two)
```

在列表4中，我使用Ruby的require方法（你可以在Kernel类中找到它）导入了dictionary模块。然后，它使用"Object.new"语法创建Word的一个新实例（ebullient）。尽管我导入了dictionary模块，但是我仍然需要限定对象实例，因此我们使用了"Dictionary::Word"限定符。

六、默认参数值

你是否注意到，在列表4中，当我创建happy_wrd实例时，我并没有一个definition或synonym的集合？我仅传递了拼写和语法成份。因为Ruby支持参数默认值，所以我成功地进行了省略。在列表3中定义的Word的initialize方法中，我指定了定义"=[]"和同义词"=[]"作为参数，这也就是向Ruby指出，如果它们不被调用者包括在内，那么将把它们默认地指定为空集合。还要注意，在列表3中，Definition的initialize方法是如何

支持默认参数的-通过把example_sentences设置为一个空集 (Ruby的nil相应于Java中的null) 来实现。在前面的列表1中 , 我必须用Java语言创建三个构造器才能取得与此相同的灵活性 ! 现在 , 请注意 , 在列表5中 , 通过使用灵活的initialize() 方法我创建了一个不同的Word实例。 列表5.Ruby的灵活性

```
! require "dictionary"
defn = Dictionary::Definition.new("Skill in or performance of tricks")
defn_two =
```

```
Dictionary::Definition.new("sleight of hand")
defs = [defn , defn_two]
tricky_wrd = Dictionary::Word.new("prestidigitation" , "noun" , defs)

```

在我定义了两个Definition以后 , 我把它们添加到一个集合(在Java语言中 , 就象一个数组)。然后 , 我把该集合传递给Word的initialize()方法。

七、集合运算

Ruby的集合运算能力也相当简单-你是否看到在Word类中的add_definition和add_synonym方法

```
? this.exampleSentences.add(exampleSentence)
```

Ruby的集合运算极其简明。在列表6中 , 你可以看到合并集合(使用操作符)和存取成员(经由[position])是多么容易 , 这样做你不需要担心任何事情。 列表6.快速的集合运算

```
require "dictionary"
idef_1 = Dictionary::Definition.new("Sad and lonely because deserted")
idef_2 = Dictionary::Definition.new("Bereft. forsaken")
defs = [idef_1 , idef_2]
idef_3 = Dictionary::Definition.new("Wretched in appearance or condition")
idef_4 = Dictionary::Definition.new("Almost hopeless. desperate")
defs_2 = [idef_3 , idef_4]
n_def = defs + defs_2
#n_def现在是[idef_1 , idef_2 , idef_3 , idef_4]
#生成idef_2
n_def[1] #生成nil
n_def[9] #生成nil
n_def[1..2] #生成[idef_2 , idef_3] 注意
```

，列表6中的代码仅涉及到Ruby中集合运算的基本内容！八、RubyBeans？你可能已经从列表3的两个加粗的类中注意到，Ruby支持使用速记符号来定义属性：它们是attr_reader和attr_writer。因为我使用了这种符号，所以我可以set和get在我的Word类中的相应的属性，见列表7：列表7.使用attr_reader和attr_writerrrequire "dictionary" wrd = Dictionary::Word.new("turpitude", "Noun") wrd.part_of_speech # "Noun" wrd.spelling # "turpitude" wrd.spelling = "bibulous" wrd.spelling # "bibulous" syns = [Dictionary::Word.new("absorptive", "Adjective"), Dictionary::Word.new("imbibing", "Noun")] # 危险! wrd.synonyms = syns = syns # 出现错误提示-"Exception: undefined method `synonyms=..." attr_reader和attr_writer都不是关键字，而实际上都是Ruby中的方法(可以在Module类中找到它们)-它们使用符号作参数。一个符号是以冒号(:)开头的任何变量，甚至在有些情况下，符号本身就是对象！注意，因为我在列表3中使synonyms成为只读的，所以Ruby否定了我在列表7中最后一行代码的尝试。另外，我可以使使用attr_accessor方法编写属性声明代码以指出一个属性既可读也可写。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com