## 

## سند زبان برنامه‌نویسی

## Jepeto

## نسخه 1.۰.1

## ۱. مقدمه

زبان Jepeto یک زبان برنامه نویسی تابعی مشابه زبان ML است که در آن type declaration وجود ندارد و Type Inference دارد. در این زبان توالی دستورات به صورت sequential نداریم و تنها میتوان فراخوانی تابع یا دستورات شرطی داشته باشیم. در واقع در این زبان دستور assignment نداریم. ساختار تکرار و دستورات حلقه نیز نداریم. این زبان قابلیت خاص anonymous function را دارد که در ادامه بیشتر توضیح داده می‌شود. این زبان از انواع داده‌ای پیش‌تعریف int، boolean و string استفاده می کند و همچنین قابلیت تعریف لیست از یکی از انواع گفته شده است (و البته لیستی از لیست نداریم).

در هر فایل از این زبان، تعدادی تابع می‌تواند وجود داشته باشد و در انتهای آن یک بخش main داریم که مشخص می‌کند برای اجرای برنامه چه تابعی با چه مقادیر اولیه‌ای فراخوانی شود (تنها یک فراخوانی در تابع main است و به تبع در آن توالی فراخوانی توابع مختلف نداریم.)

در این زبان، کد برنامه درون یک فایل با پسوند jp. قرار دارد.

## ۲.ساختار کلی

یک برنامه به زبان Jepeto از قسمت های زیر تشکیل شده است:

* تعریف تعدادی تابع
* یک بخش main

یک نمونه از کد در این زبان به صورت زیر است:

|  |
| --- |
| func testMethod(test1, test2, i):  if i not 10:  return testMethod (test1, test2::i, i+1);  else  if test1 is test2[i]:   return true;  else:  return false;    main: print(testMethod(10,[],0)); |
|  |

## ۲-۱. قواعد کلی نحو

زبان Jepeto به بزرگ و کوچک بودن حروف حساس است. در این زبان، وجود کاراکتر‌های Space ، Tab و خط جدید تاثیری در خروجی برنامه ندارند. جزئیات مربوط به Scope ها و خطوط برنامه در ادامه به طور مفصل توضیح داده‌ خواهد شد.

## ۲-۲. کامنت‌ها

در این زبان، کامنت‌ها تنها تک‌خطی هستند و تمامی کاراکتر‌های بعد از # تا انتهای خط کامنت به حساب می‌آیند.

|  |
| --- |
| func getLastElement(L):return L[L.size-1];main: print getLastElement([“A”,”B”,”C”]) # This is a comment! |
|  |

## 

## ۲-۳. قواعد نام‌گذاری توابع و آرگومان‌ها

اسامی انتخابی برای نام‌گذاری توابع و آرگومانها باید از قواعد زیر پیروی کنند:

* تنها از کاراکتر‌های a..z،A..Z،‌ ــ و ارقام تشکیل شده باشند.
* با رقم شروع نشوند.
* معادل کلید‌واژه‌ها نباشند. در جدول زیر تمامی کلیدواژه‌های زبان Jepeto آمده‌است:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| boolean | string | int | and | func |
| or | if | print | is | return |
|  | false | true | not | else |

* نام هر تابع، یکتاست.
* نام هر آرگومان در یک Scope یکتاست.

## ۳. بخش Main و نحوه فراخوانی توابع

بخش main

همه توابع باید مقدار بازگشتی داشته باشند. همچنین، تعریف یک متد به صورت زیر انجام می‌شود:

|  |
| --- |
| func method(arg1, arg2, arg3, …):  # bodyreturn true; |
|  |

فراخوانی متد‌ها یک Statement و Statement دیگر این زبان return است. در انتهای Statement یک کاراکتر ; نوشته می‌شود. برای تعریف بلاک از {} استفاده می شود.

در بخش main تنها می‌توان یک تابع را فراخوانی کرد.

در این زبان، فراخوانی متد به صورت call-by-reference است.

## ۳. نحوه تعریف توابع و توابع گمنام

در کد زیر maxfinder یک تابع معمولی و equal یک تابع گمنام است که بدون اسم گذاری در فراخوانی یک تابع تعریف شده است و تنها در همان تابع قابل فراخوانی است.

|  |
| --- |
| func maxfinder(arr,i,equal,max):  if arr.size < i:  return max;  else :  maxfinder(arr,i+1,equal,equal(arr[i],arr[i+1]));    main: maxfinder(arr=[1,2,3,4,5],i=0,(a,b)->{if a>b: return a; else: b; },max=-1); |
|  |

## ۵. انواع داده

در زبان Jepeto، سه تایپ پایه string ،int و boolean وجود دارند.

در این زبان، میتوان لیست هم به عنوان ورودی توابع تعریف شود که میتواند int list یا string list و یا boolean list باشد.

## ۶. آرگومان‌ها

آرگومان‌های تابع می‌تواند از صفر تا چند مورد باشد که از هر یک از انواع پیش تعریف و یا لیست و یا یک تابع دیگر باشد. با فراخوانی یک تابع با مقادیر ورودی مشخص، آرگومان‌های آن را مقداردهی می‌کنیم.

در صورتی که به آرگومانی مقداری نسبت داده نشود، مقدار آن برابر با مقدار پیش‌فرض تایپ خود در نظر گرفته می‌شود. مقادیر پیش‌فرض تایپ‌های مختلف در جدول زیر آمده‌است:

|  |  |
| --- | --- |
| 0 | int |
| false | boolean |
| "" | string |

تایپ‌های موجود در این زبان در جدول زیر آمده‌است:

|  |
| --- |
| int |
| String |
| boolean |
| list |

در این زبان، تایپ‌های int، string و boolean از نوع primitive هستند.(خود مقادیر در آن‌ها ذخیره می‌شوند نه اشاره‌گری به خانه‌ای از حافظه) و تایپ‌های دیگر، از نوع non-primitive هستند و در آن‌ها اشاره‌گری به خانه‌ای از حافظه وجود دارد.

## ۷. عملگر‌ها

عملگرها در زبان Jepeto به چهار دسته‌ی عملگرهای حسابی، مقایسه‌ای، منطقی، عملگر تخصیص تقسیم می‌شوند.

## 

## ۷-۱.عملگرهای حسابی

این دسته از عملگرها تنها روی اعداد عمل می‌کنند، لیست این عملگرها در جدول زیر آمده است. در مثال‌های استفاده شده A برابر 20 و B را برابر 10 در نظر بگیرید:

|  |  |  |  |
| --- | --- | --- | --- |
| **عملگر** | **شرکت‌پذیری** | **توضیح** | **مثال** |
| + | چپ | جمع | A+B=30 |
| - | چپ | تفریق | A-B=10 |
| \* | چپ | ضرب | A\*B=200 |
| / | چپ | تقسیم | A/B=2  B/A=0 |
| - | راست | منفی تک‌عمل‌وندی | -A=-20 |

## ۷-۲. عملگرهای مقایسه‌ای

این عملگرها وظیفه‌ی مقایسه را دارند، پس نتیجه‌ی آن‌ها باید مقدار صحیح یا غلط (true, false) باشد. با این حساب خروجی این عملگرها یک boolean است.

توجه داشته باشید که عملوند عملگرهای < و > تنها از جنس عدد صحیح هستند. همچنین برای عملگرهای is و not نیز باید تایپ عملوندها یکسان باشند و در صورت آرایه بودن، اندازه‌ی آن‌ها نیز برابر باشد؛ در غیر اینصورت باید خطای کامپایل گرفته شود.

لیست عملگرهای مقایسه‌ای در جدول زیر آمده است. در مثال‌های استفاده شده مقدار A را برابر 20 و مقدار B را برابر 10 بگیرید:

|  |  |  |  |
| --- | --- | --- | --- |
| **عملگر** | **شرکت‌پذیری** | **توضیح** | **مثال** |
| is | چپ | تساوی | (A is B) = false |
| not | چپ | عدم تساوی | (A not B) = true |
| > | چپ | کوچکتر | (A < B) = false |
| < | چپ | بزرگتر | (A > B) = true |

## ۷-۳. عملگرهای منطقی

در زبان Jepeto عملیات منطقی تنها روی نوع داده‌ی Bool قابل اعمال است.این عملگرها در جدول زیر لیست شده‌اند. در مثال‌های استفاده شده A را برابر true و B را برابر false در نظر بگیرید:

|  |  |  |  |
| --- | --- | --- | --- |
| **عملگر** | **شرکت‌پذیری** | **توضیح** | **مثال** |
| and | چپ | عطف منطقی | (A and B) = false |
| or | چپ | فصل منطقی | (A or B) = true |
| ~ | راست | نقیض منطقی | (~A) = false |

## ۷-۵. اولویت عملگرها

اولویت عملگرها طبق جدول زیر است:

|  |  |  |  |
| --- | --- | --- | --- |
| **اولویت** | **دسته** | **عملگرها** | **شرکت‌پذیری** |
|  | برای سایزلیست | . |  |
|  | برای اضافه کردن به لیست | :: | چپ |
| **۱** | پرانتز | () | چپ |
| **۲** | تعریف لیست | [] | چپ |
| **۳** | تک عملوندی | - ~ | راست |
| **۴** | ضرب و تقسیم | \* / | چپ |
| **۵** | جمع و تفریق | +- | چپ |
| **۶** | رابطه‌ای | < > | چپ |
| **۷** | مقایسه‌ی تساوی | is not | چپ |
| **۸** | عطف منطقی | and | چپ |
| **۹** | فصل منطقی | or | چپ |
| **10** | کاما(ورودی متدها) | , | چپ به راست |

## ۸. ساختار تصمیم‌گیری

در زبان Jepeto تنها ساختار تصمیم‌گیری، if… else است:‌

|  |
| --- |
| if a is 2:   return true;else:  return false; |

همچنین ساختار if می‌تواند بدون else استفاده گردد.

## ۱۰. قوانین Scopeها

## ۱۰-۱ Scopeهای موجود در زبان

به طور کلی در زبان Jepeto موارد زیر در اسکوپ جدیدی قرار دارند:‌

1- آرگومان‌ها و خطوط کد داخل یک متد.

2- Expression مورد بررسی ساختار if و else و خطوط کد داخل آن

## ۱۰-۲. قوانین Scopeها

نکات زیر در مورد Scopeها وجود دارد:‌

* متغیرهایی که داخل یک تابع تعریف می‌شوند در Scopeهای بیرون آن دسترس‌پذیر نیستند و صرفاً در Scopeهای درون آن قابل دسترسی هستند.
* امکان تعریف آرگومان با نام یکسان در یک Scope وجود ندارد.

## ۱۱. توابع ‌و فیلدهای پیش‌فرض

در زبان Jepeto، تنها یک تابع پیش‌فرض وجود دارد و آن، تابع print که ورودی از هر نوع که باشد چاپ می‌کند.

## ۱۱-۱. تابع print

این تابع به صورت ضمنی تعریف شده است و می‌تواند یک آرایه‌(با هر طولی) و یا یک مقدار int یا string دریافت کند و آن را در کنسول چاپ کند. نمونه‌ای از این دستور به صورت زیر است:

|  |
| --- |
| print("Hello Programmer!") |

## ۱۱-۲. فیلد size

این فیلد برای لیستها تعریف می‌شود و طول یک لیست را بازمی‌گرداند. به عنوان مثال:

|  |
| --- |
| func length(arr):  return arr.size;    main: print(length([1,2,3]); # print 3 |

## ۱۲. مثال‌ها

در ادامه، چندین کد نمونه از این زبان آمده‌است:

|  |
| --- |
| func Multiply(a,b):  if a is 0:return a+Multiply(a-1,b);   main: print(Multiply(4,5)); |