

Documentation for lab

Description for Scanner Class

Scanner():

description: default constructor, initialize the pif and symbol table

in: NA

out: NA

Scanner(List<Token> pif, SymbolTable symbolTable):

description: default constructor, initialize the pif and symbol table
with the given params

in: NA

out: NA

scan(String filename)

description: it parses the file and construct the PIF and SymbolTable

in: filename of type String

out: NA

isConstant(String token):

description: checks if a token is constant according to a predefined rule

in: token of type String

out: true if it according to the rule, false otherwise

isIdentifier(String token):

description: checks if a token is an identifier according to a predefined rule

in: token of type String

out: true if it according to the rule, false otherwise

isReservedWord(String token)

description: checks if a token is a reserved word

in: token of type String

out: true if it according to the rule, false otherwise

isSeparator(String token), isSeparator(char token)

description: checks if a token is a separator

in: token of type String

out: true if it according to the rule, false otherwise

isOperator(String token)

description: checks if a token is an operator

in: token of type String

out: true if it according to the rule, false otherwise

mightBeOperator(String token)

description: checks if a token is partially an operator

in: token of type String

out: true if it according to the rule, false otherwise

split(String code)

description: splits the code text into tokens

in: code of type String

out: List<String> - list of tokens

===== p1 =====
=====

```
int a = input()
int b = input()
int c = input()

if (a >= b and a >= c) {
    print("max is ")
    print(c)
} elif (b >= a and b >= c) {
    print("max is")
    print(b)
} else {
    print("max is", "hello me")
    print(c)
}
print('e')
```

===== o1 =====
=====

```
PIF
int -- -1
id -- 97
= -- -1
input -- -1
( -- -1
) -- -1
int -- -1
id -- 98
= -- -1
input -- -1
( -- -1
) -- -1
int -- -1
id -- 99
= -- -1
input -- -1
( -- -1
) -- -1
if -- -1
( -- -1
id -- 97
>= -- -1
id -- 98
```

```
and -- -1
id -- 97
>= -- -1
id -- 99
) -- -1
{ -- -1
print -- -1
( -- -1
const -- 78
) -- -1
print -- -1
( -- -1
id -- 99
) -- -1
} -- -1
elif -- -1
( -- -1
id -- 98
>= -- -1
id -- 97
and -- -1
id -- 98
>= -- -1
id -- 99
) -- -1
{ -- -1
print -- -1
( -- -1
const -- 46
) -- -1
print -- -1
( -- -1
id -- 98
) -- -1
} -- -1
else -- -1
{ -- -1
print -- -1
( -- -1
const -- 46
, -- -1
const -- 42
) -- -1
print -- -1
( -- -1
id -- 99
) -- -1
} -- -1
print -- -1
( -- -1
' -- -1
id -- 1
' -- -1
) -- -1
```

symbolTable

```
1  - [e]
42 - ["hello me"]
46 - ["max is"]
78 - ["max is "]
97 - [a]
98 - [b]
99 - [c]
```

===== p2 =====

=====

```
int[] lst = [1, 2, 3, 4, 5]
int n = 5
int maxi = lst[1]
```

```
for int i in (1:n) {
    maxi = maxi < lst[i] ? lst[i] : maxi
}
```

```
print("max is ")
print(maxi)
```

===== o2 =====

=====

PIF

```
int -- -1
[ -- -1
] -- -1
id -- 39
= -- -1
[ -- -1
const -- 49
, -- -1
const -- 50
, -- -1
const -- 51
, -- -1
const -- 52
, -- -1
const -- 53
] -- -1
int -- -1
id -- 10
= -- -1
const -- 53
int -- -1
id -- 31
= -- -1
id -- 39
[ -- -1
const -- 49
] -- -1
for -- -1
int -- -1
```

```

id -- 5
in -- -1
( -- -1
const -- 49
: -- -1
id -- 10
) -- -1
{ -- -1
id -- 31
= -- -1
id -- 31
< -- -1
id -- 39
[ -- -1
id -- 5
] -- -1
? -- -1
id -- 39
[ -- -1
id -- 5
] -- -1
: -- -1
id -- 31
} -- -1
print -- -1
( -- -1
const -- 78
) -- -1
print -- -1
( -- -1
id -- 31
) -- -1

```

```

symbolTable
5 - [i]
10 - [n]
31 - [maxi]
39 - [lst]
49 - [1]
50 - [2]
51 - [3]
52 - [4]
53 - [5]
78 - ["max is "]

```

===== p3 =====

```

string[] lst = ["hello", "world", "a!"]
int n = 3
string rez

```

```

for int i in (0:n) {
    rez += lst[i]
}

```

```
rez += ''
}
```

```
print(rez)
```

o3

PIF

```
string -- -1
```

```
[ -- -1
```

```
] -- -1
```

```
id -- 39
```

```
= -- -1
```

```
[ -- -1
```

```
const -- 0
```

```
, -- -1
```

```
const -- 20
```

```
, -- -1
```

```
const -- 98
```

```
] -- -1
```

```
int -- -1
```

```
id -- 10
```

```
= -- -1
```

```
const -- 51
```

```
string -- -1
```

```
id -- 37
```

```
for -- -1
```

```
int -- -1
```

```
id -- 5
```

```
in -- -1
```

```
( -- -1
```

```
const -- 48
```

```
: -- -1
```

```
id -- 10
```

```
) -- -1
```

```
{ -- -1
```

```
id -- 37
```

```
+= -- -1
```

```
id -- 39
```

```
[ -- -1
```

```
id -- 5
```

```
] -- -1
```

```
id -- 37
```

```
+= -- -1
```

```
' -- -1
```

```
' -- -1
```

```
} -- -1
```

```
print -- -1
```

```
( -- -1
```

```
id -- 37
```

```
) -- -1
```

symbolTable

```
0 - ["hello"]
```

```
5 - [i]
10 - [n]
20 - ["world"]
37 - [rez]
39 - [lst]
48 - [0]
51 - [3]
98 - ["a!"]
```

===== pe =====

```
string[] lst = ["hello", "world", "!]
int n = 3
string 222rez
```

```
for int i in 0:n {
    rez += lst[i]
    rez += ' '
}
```

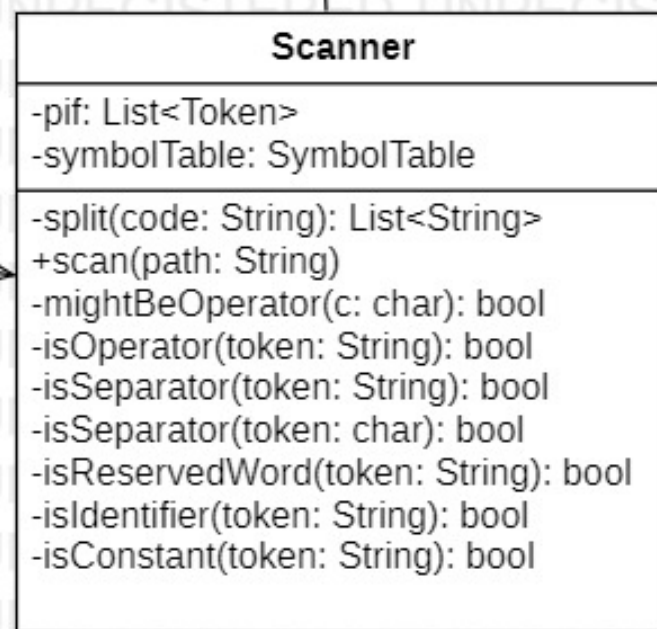
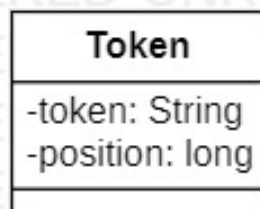
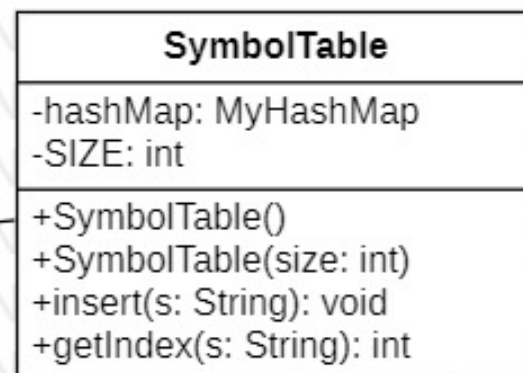
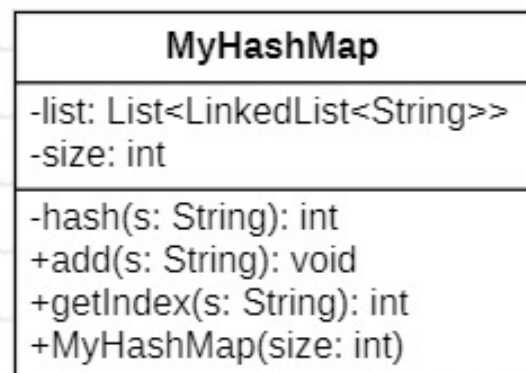
```
print(rez)
```

===== oe =====

```
PIF
Cannot classify token: 222rez
string -- -1
[ -- -1
] -- -1
id -- 39
= -- -1
[ -- -1
const -- 0
, -- -1
const -- 20
, -- -1
' -- -1
] -- -1
int -- -1
id -- 10
= -- -1
const -- 51
string -- -1
for -- -1
int -- -1
id -- 5
in -- -1
const -- 48
: -- -1
id -- 10
{ -- -1
id -- 37
+= -- -1
id -- 39
```

```
[ -- -1
id -- 5
] -- -1
id -- 37
+= -- -1
' -- -1
' -- -1
} -- -1
print -- -1
( -- -1
id -- 37
) -- -1
```

```
symbolTable
0 - ["hello"]
5 - [i]
10 - [n]
20 - ["world"]
37 - [rez]
39 - [lst]
48 - [0]
51 - [3]
```

+has a