# **PASS** - Computer Systems

Week 11

# 1. Variable types and scopes

Can you find all static, field, local and parameter variables in the code below?

```
class BankAccount {
    // Class variables
    static int nAccounts;
    static int bankCommission;
    // account properties
    field int id;
    field String owner;
    field int balance;

method void transfer(int sum, BankAccount from, Date when) {
      var int i, j; // Some local variables
      var Date due; // Date is a user-defined type
      let balance = (balance + sum) - commission(sum * 5);
      // More code ...
}
```

type	variable	declared in	scope	memory segment
Static	n Accounts, pank Commision	class	class	Statiz
field	id, owner, balance	clasz	methods can access	this
(0 Ca)	i,i, due	Subloutine	Suplontine	101
parameter	rum, from,	subvoutine varameter vist	Sniturrduz	arg

#### 2. Write a Player class that contains:

- 3 filed variables: string name, int age, int level
- 1 static variable: string color
- The constructor new() takes in 3 parameters: name\_, age\_. The constructor sets the object's name and age accordingly, and sets level = 0.
- A void function change\_color(string color\_) will set the static variable color to be color\_
- A method upgrade() increases the level by 1, and return the player's updated level.
- How would you call new(), change\_color(), and upgrade()?

```
class Player &
Static String color
field String name
field string name
field int lase

Constructor Player new (String name-, int age-) & do new. Change-Colon(M)

Constructor Player new (String name-, int age-) & do new. Change-Colon(M)

let name = name-
let age = age-
let level = 0

return this

function void change-colour (string colour-) &

let color = colour-

return;

method int uprade() {

let level = level t l

yetrum level

Y
```

#### 3. Tokenizer

JACK program --> tokenizer --> parser --> code generator --> VM code

tokenizer and parser form syntax analyzer

tokenizer, parser and code generator form compiler

Tokenizer: next\_token(), have(TokenKind), mustbe(TokenKind)...

a. Consider these rules for a tokenizer. What tokens would be produced from the following string assuming that whitespace is not returned as tokens?

"23 hobbits50 99chickens"

```
int ::= '0' | (('1'-'9')('0'-'9')*)
name ::= ('a'-'z') ( 'a'-'z'|'0'-'9' )*
```

Answer:

```
1. 23 -int
2. hobbits 50 - name
3. 99. -int
4- chickens hame
```

b. Consider these rules for a tokenizer. What tokens would be produced from the following string assuming that whitespace is not returned as tokens?

"09.06.0.0.0"

```
num ::= ('0' | (('1'-'9')('0'-'9')*)) ('.' ('0'-'9')
*)?
dot ::= '.'
```

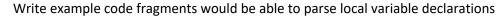
Answer:

4. Why is JACK's grammar LL(1)?

only I token is needed to determine the rule.

- 5. Parser: parse\_class(), parse(type), parse
  - a. Consider the grammar rule for a Jack local variable declaration(e.g. var int x,y;):

```
varDec ::= 'var' type varname (',' varname)* ';'
```



6. Construct class-wise and method-wise symbol table for code below:

```
class BankAccount
{
    static int nAccounts;
    static int bankCommission;
    field int id;
    field string owner;
    field int balance;
    method void transfer(int sum, BankAccount from, Date whe
n)
    {
        var int i, j;
        var Date due;

        let balance = (balance + sum) - commission(sum *
5);
    }
}
```

Question: how many symbol tables exist in a class? – A class wise symbol table + one symbol table for each subroutine. Commonly implemented as a linked list.

## class wise:

name	type	Kind	offset
n accounts	int	static	0
banl2commision	int	Stat Ic	1
id	int	This	D
owner	strin 9	this	1
valance	int	this	2

## method-wise:

name	type	Kind	offset
Swm	int	arg	1
From	BankAccount	ara	2
when	Dare	ora	3
Bank Account	this	ara	0
Ì	fn f	10261	D
7	int	loca)	)
due	Date	local	2

7. Construct class-wise and method-wise symbol table for code below:

```
class Player
     {
         field String name;
         field int age, level;
         static String color;
         static int color_intesity;
         constructor Player new(String name_, int age_) {
             let name = name_;
10
             let age = age_;
             let level = 0;
12
             return this;
         function void change_color(String color_, int intensity) {
             var int i;
             let i = intensity;
             while (i > 0)
                 let color_intesity = color_intesity * 3;
21
                 let i = i - 1;
             let color = color_;
             return;
         method int upgrade() {
             let level = level + 1;
             return level;
```

class wise:

name	type	Kind	offset
name	String	field	0
age	int	field	(
Icvel	int	field	2
color	String	Static	0
color-intensity	int	Static	

#### constructor wise:

name	type	Kind	offset
name -	String	arq	O
age -	int	out	
u —		. 1	

#### function-wise:

name	type	Kind	offset
color_	String int	ayoument	Ó
color_ intensity_	int	orgument local	1
i	int	10ca1	0

## method-wise:

name	type	Kind	offset
this	Player	argument	0
	U	U	