

accept ()  $\rightarrow$   $\begin{cases} \text{accept current token} \\ \text{nextToken()}. \end{cases}$

nextToken ()

There is a case for calling accept (...) even when nextToken() would be adequate:  
 accept(...) would do a redundant check in such cases, but the advantage is that changes to the code that are implemented later would be less likely to break the code.

## TinyJ Virtual Machine (VM)

The memory of the TinyJ VM consists of 2 disjoint regions with separate address spaces:

- ① code memory
- ② Data memory

code memory is used to store instructions that can be fetched for execution

[ For assignment 2, there will be no fetching of instructions for execution, your program will just compile the TinyJ source file into a sequence of TinyJ VM instructions ]

Data memory is used to store:

- values of variables.
- contents of arrays.
- characters of string literals

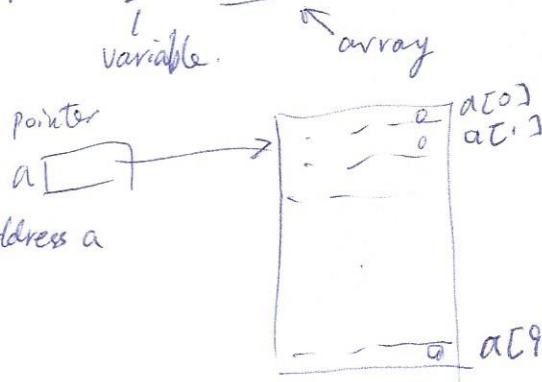
Data memory address a is unrelated to code memory address a

Data memory consists of 3 ~~re~~ regions:

a) A statically allocated region (used to store values of static variables and characters of string literals)

\* In TinyJ, (unlike Java) strings are not objects, Arrays are the only objects in TinyJ

int[] a = new int[100];  
                   ↑  
                   Variable.



b) a stack-dynamically allocated region, used to store values of local variables of methods (including ~~formal~~ formal parameters of methods)

c) A heap-dynamically allocated region, used to store arrays (which are the objects in TinyJ)

