Jasmin Instruction Part 4

Csc3210
Yuan Long
Array

- Create an integer array
- Storing a data into an integer array
- Accessing a value in an integer array
- Create any type array
- Create multi-dimensional array
- Get the length
Create an integer array

- Make a new array with 10 elements.

  \[
  \text{int[]} \text{ localVariable2=\text{new int[10]};}
  \]

- Use local variable 2 to refer to it.

  \[
  \text{bipush 10} \quad ; \text{number of elements} \\
  \text{newarray int} \quad ; \text{make the array and return} \\
  \text{astore_2} \quad \quad \quad ; \text{the array address on top of stack} \\
  \text{astore_2} \quad \quad \quad ; \text{put the array address} \\
  \text{astore_2} \quad \quad \quad ; \text{in localVariable2}
  \]
Create an integer array

Syntax:

```
newarray basic Types in {I,J,F,D,B,S,C,Z} ;
```

You can use:

```
newarray int
newarray long
newarray float
newarray double
newarray byte
newarray short
newarray char
newarray boolean
```
Storing a data into an integer array

localVariable2[localVariable1] = 3

• push localVariable2 as address

• push localVariable1 as integer (index)

• push 3

• localVariable2[localVariable1] = stack top
aload_2     ; localVariable2 is the array address
iload_1     ; use localVariable1’s value as the index
iconst_3    ; the integer value to store
iastore     ; array[localVariable1] = 3

Note: iastore pops an int from the stack and stores it in an array of ints. It is different from astore_X.
How the stack works for iastore?

localVariable2[LocalVariable1]=3
Accessing a value in an integer array

```plaintext
localVariable3 = localVariable2[5]

• push localVariable2 as address
• push 5 as integer (index)
• localVariable3 = localVariable2[5]
```
aload_2 ; localVariable2 is the array address
iconst_5 ; the index is the number 5
iaload ; access array[5]
istore_3 ; put array[5] in localVariable3

Note: Retrieve an entry from int array and places it in an stack.
How the stack works for iaload?

|-------------------------|--------------|

Load the value of localVar2[5]
Example 1

Create the array, of 100 integers.

bipush 100
newarray int ; int[] a=new int[100]

The array address will now be on the stack.

dup ; Duplicate the last thing on the stack, which is a reference to the array.
icontst_0 ; push 0 as the index (assuming i=0)
icontst_1 ; push 1 as the value to store
iastore ; pop the last 3 things, and ; store the value at index in the integer array ; a[0] = 1

t[ ] a=new int[100];
a[0]=1;
a[1]=2;
a[3]=3;
The array address will still be on the stack, since we did not remove it.

We repeat this with different indices and values. Later, we store the array address into a local variable.

    astore_1 ; put the array address into localVariable1
Create any type array

• Make a new array with 1000 Strings.
String localVariable2=new String[1000];

• Use local variable 2 to refer to it.

```
ldc 1000 ; number of elements
anewarray java/lang/String ; make the array
astore_2 ; put the array address
    ; localVariable2
```
Create multi-dimensional array

• Make a new array below.

float[][][][] localVariable2 = new float[6][5][3];

• Use local variable 2 to refer to it.

  bipush 6 ; size of array in 3rd dimension = 6
  bipush 5 ; size of array in 3rd dimension = 5
  bipush 3 ; size of array in 3rd dimension = 3
  Multianewarray [[[F 3; create a 6*5*3 array of floats
Get the length

```java
int size = localVar2.length

aload_2    ; load previously defined localVar2 array
           ; (localVar2[10])
arraylength ; pop array address, push its length
istore_1   ; size = array length
```

How about the length for a multi-dimensional array?
<table>
<thead>
<tr>
<th>Array Element Type</th>
<th>Store Operation</th>
<th>Load Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>iastore</td>
<td>iaload</td>
</tr>
<tr>
<td>long</td>
<td>lastore</td>
<td>laload</td>
</tr>
<tr>
<td>double</td>
<td>dastore</td>
<td>daload</td>
</tr>
<tr>
<td>float</td>
<td>fastore</td>
<td>faload</td>
</tr>
<tr>
<td>char</td>
<td>castore</td>
<td>caload</td>
</tr>
<tr>
<td>short</td>
<td>sastore</td>
<td>saload</td>
</tr>
<tr>
<td>byte</td>
<td>bastore</td>
<td>baload</td>
</tr>
<tr>
<td>boolean</td>
<td>bastore</td>
<td>baload</td>
</tr>
<tr>
<td>array (address)</td>
<td>aastore</td>
<td>baload</td>
</tr>
<tr>
<td>object (address)</td>
<td>aastore</td>
<td>aaload</td>
</tr>
</tbody>
</table>

**Table**: Array element type and corresponding store & load operation
Example 2:

Int[] a=new int[5];

Int[][] b=new int[2][3];

String[] c=new String[256];

String[][] d=new String[256][5];
Example 2:

```
a[3]=-1;

b[1][2]=1;

b[1][2]=a[3];

c[3]="string1";

d[1][2]=c[3];
```
Practice

• Bytecode file
  – reads 3 integers and do summation.
  – However, for more than 3 integers, it does not work.
  – Now I want to get the sum of 5 integers.

• Solve that problem by modifying the condition in for loop for bytecode file.

```java
int sum=0;
for(int i=0;i<3;i++){
    Scanner s=new Scanner(System.in);
    int number=s.nextInt();
    sum+=number;
}
```
```java
int sum=0;
for(int i=0;i<5;i++){
    Scanner s=new Scanner(System.in);
    int number=s.nextInt();
    sum+=number;
}
```
How?

• Locate the place of number 3?
  xxd sum.j | grep 03  

• Locate the place for if_icmp, if**?
  xxd practice.j | grep 10 | grep 0xXX  
XX is the OpCode for corresponding instructions.
Program challenge

- Initialize array with length n provided by user.
- Array size is too small, program has problems when input number n is greater than 100.
- Locate the number for initializing array size.
Homework

• Use array to finish program challenge 6
• Storing each Fibonacci number in an array.
• Write a method which can print out values in an array.