Module 5

Arrays

Objectives

- Declare and create arrays of primitive, class, or array types
- Initialize the elements of an array
- Determine the number of elements in an array: *array.length*
- Create a multidimensional array
- Copy array values from one array to another
- Comparing ,Sorting,Searching an Array
- *Enum* Types

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Declaring and Creating Arrays

- An array is an *object*; it groups data objects of the *same type*.
- Declare arrays of *primitive* or *class* types, just create space for a reference.

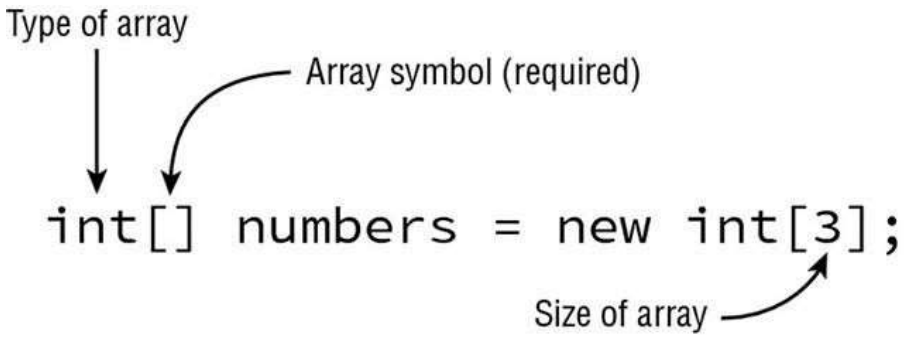
```
char s[];
Point p[];

char[] s;
Point[] p;
```

As parameter, also allowed:

```
char... s;
Point... p;
```


Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

- Use the *new* operator to create an array
- Access each individual value through an integer index.



Copyright 2007-2020, All Rights Reserved. Chen Xudong, School of Software Engineering of BJTU

Creating Arrays

For example, a primitive (`char`) array:

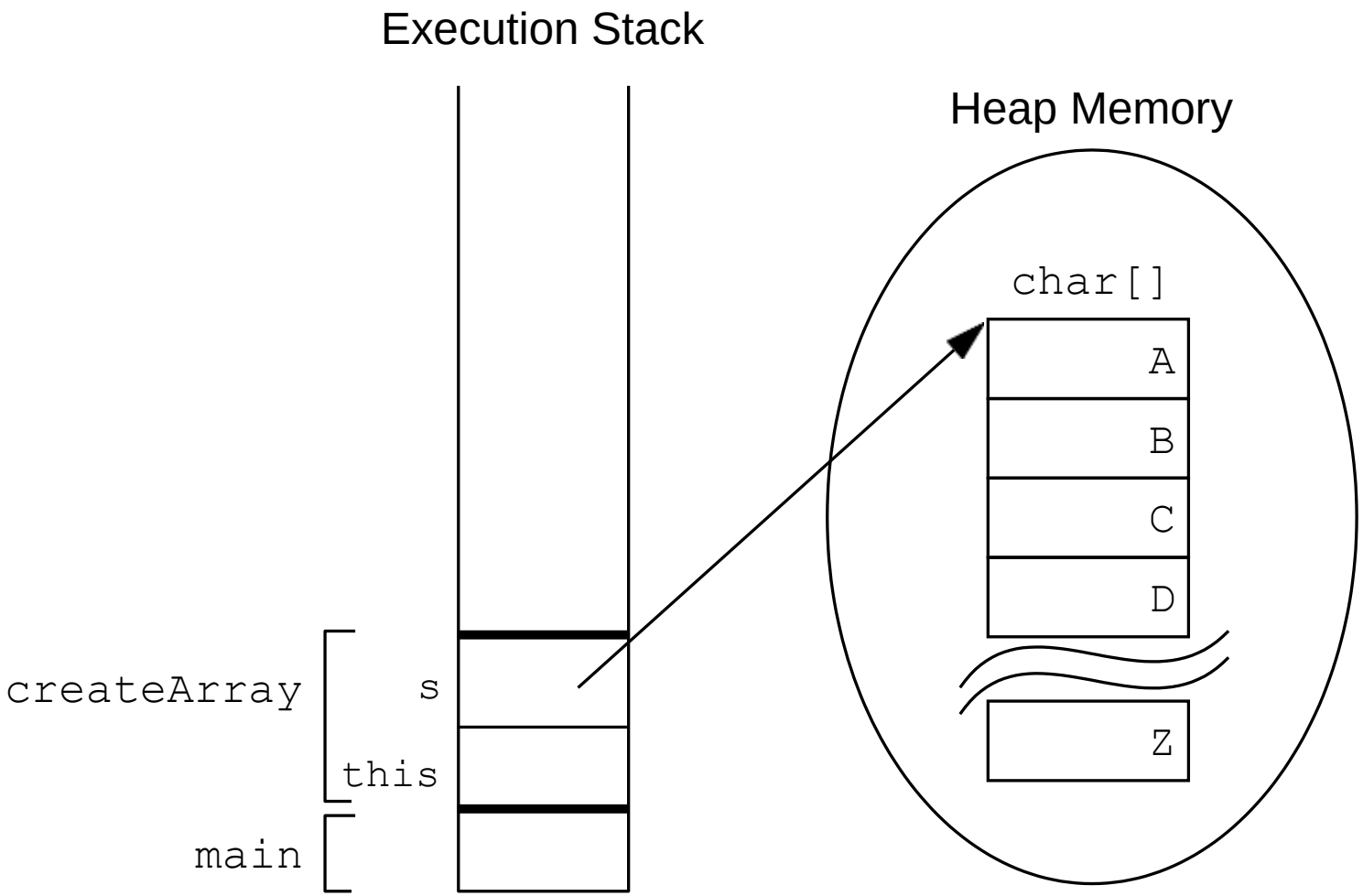
```
1   public char[] createArray()
2     { char[] s;
3
4     s = new char[26];
5     for ( int i=0; i<26; i++ )
6       { s[i] = (char) ('A' + i);
7     }
8
9     return s;
10  }
```

To find the number of elements of an array, use *array*.*length*:

```
int[] a = new int[100];
for (int i = 0; i < a.length; i++)
    System.out.println(a[i]);
```


Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Creating a Primitive Array



Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

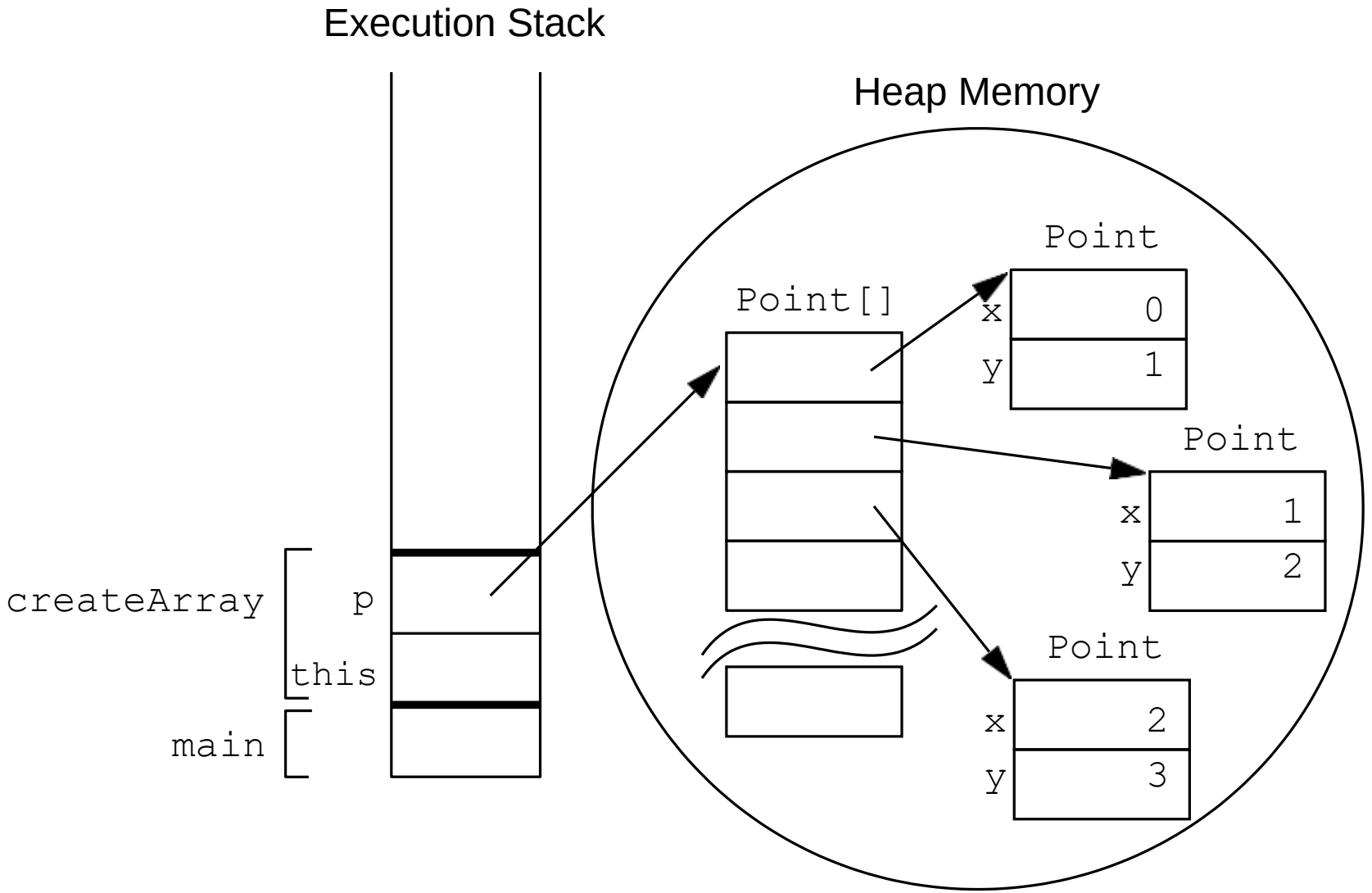
Creating aReference Array

Another example, an object array:

```
1   public Point[] createArray()
2     { Point[] p;
3
4     p = new Point[10];
5     for ( int i=0; i<p.length; i++ ) {
6       p[i] = new Point(i, i+1);
7     }
8
9     return p;
10  }
```


Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Creating an Array of `Point` Objects



Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Using the Enhanced `for` Loop

- Since Java 5.0 , for iterating over arrays:

```
public void printElements(int[] list) {
  for ( int element : list ) {
    System.out.println(element);
  }
}
```

The `for` loop can be read as *for each* `element` *in* `list`

- Just print all values of an array: *Arrays.toString(a)*

  - returns a string containing the array elements, enclosed in brackets and separated by commas

  - such as "[2, 3, 5, 7, 11, 13]".

  - `System.out.println(Arrays.toString(list));`

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Initializing Arrays

- Initialize an array element.
- Create an array with initial values.

```
String[] names;
names = new String[3];
names[0] = "Georgianna";
names[1] = "Jen";
names[2] = "Simon";
```

```
String[] names = {
    "Georgianna",
    "Jen",
    "Simon"
};
```

```
MyDate[] dates;
dates = new MyDate[3];
dates[0] = new MyDate(22, 7, 1964);
dates[1] = new MyDate(1, 1, 2000);
dates[2] = new MyDate(22, 12,1964);
```

```
MyDate[] dates = {
    new MyDate(22, 7, 1964),
    new MyDate(1, 1, 2000),
    new MyDate(22, 12, 1964)
};
```


Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Array Resizing

- You cannot resize an array.
- You can use the same reference variable to refer to an entirely new array, such as:

```
int[] myArray = new int[6];
myArray = new int[10];
```


Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Copying Arrays

- Assign one array variable into another, both variables refer to the same array

```
int[] luckyNumbers = smallPrimes;
luckyNumbers[5] = 12; // now smallPrimes[5] is also 12
```

- The *System.arraycopy()* method to copy arrays :

```
System.arraycopy(from, fromIndex, to, toIndex, count);
```

```
1   //original array
2   int[] myArray = { 1, 2, 3, 4, 5, 6 };
3
4   // new larger array
5   int[] hold = { 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 };
6
7   // copy all of the myArray array to the hold
8   // array, starting with the 0th index
9   System.arraycopy(myArray, 0, hold, 0, myArray.length);
```


Copyright Copyright 2007-2020,

Command-Line Parameters

Every Java program has a *main* method with a *String[]* args parameter, indicates that the *main* method receives an array of strings, namely, the arguments specified on the command line.

```
public class Message{
  public static void main(String[] args){
    if (args[0].equals(”-h”))
      System.out.print(”Hello,”);
    else if (args[0].equals(”-g”))
      System.out.print(”Goodbye,”);
    // print the other command-line arguments
    for (int i = 1; i < args.length; i++)
        System.out.print(” “ + args[i]);
    System.out.println(”!”);
  }
}
```


Copyright Copyright 2007-2020,

Multidimensional Arrays

Arrays of arrays:

```
int[][] twoDim = new int[4][];
twoDim[0] = new int[5];
twoDim[1] = new int[5];

int[][] twoDim = new int[][4]; // illegal
```


Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Multidimensional Arrays

- Non-rectangular arrays of arrays:

```
twoDim[0] = new int[2];
twoDim[1] = new int[4];
twoDim[2] = new int[6];
twoDim[3] = new int[8];
```

- Array of four arrays of five integers each:

```
int[][] twoDim = new int[4][5];
```

- print all values of an multidimensional array:
  Arrays.deepToString(arrayName)

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Comparing Arrays

- java.util.Arrays
- Arrays.equals()
  - static boolean equals(primType[] a1, primTypeObject[] a2);
  - static boolean equals(Object[] a1, Object[] a2);

- ComparingArrays.java

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Sorting Arrays

- Arrays.sort()
  - static void sort(primType[] a);

  Sorts the specified array into ascending numerical order.
  - static void sort(primType[] a, int from, int to);

  Sorts the specified range of the array into ascending order. The extends from the index from, inclusive, to the index to, exclusive

- Lottery.java

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Sorting Arrays

- ***Comparable*** interface imposes a total ordering on the objects of each class that implements it. This ordering is referred to as the class's *natural ordering*, and the class's *compareTo* method is referred to as its natural comparison method.
- Arrays.sort()
  - staticvoidsort(Object[]a)
    Sorts the specified array of objects into ascending order, according to the natural ordering of its elements.
  - staticvoidsort(Object[]a, intfrom, intto)
    Sorts the specified range of the specified array of objects into ascendingorder, according to the natural ordering of its elements.
- SortingString.java

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Searching Arrays

- Arrays.binarySearch()
  - Searches the specified array for the specified key using the binary search algorithm.
  - SearchingArray.java
    //Search from an array of primitive type
  - SearchingString.java
    //Search from an array of object type

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Enumeration

enum Types

- A special data type that enables for a variable to be a set of predefined constants.
- The variable must be equal to one of the values that have been predefined for it

```
public enum Day {
    SUNDAY, MONDAY, TUESDAY, WEDNESDAY,
    THURSDAY, FRIDAY, SATURDAY
}
```

- Days.java TryEnumeration.java
- TestOpCodeEnum.java

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Advanced Enumerated Types

Enumerated types can have attributes and methods:

```
1   package cards.domain;
2
3   public enum Suit {
4     SPADES   (“Spades”),
5     HEARTS   (“Hearts”),
6     CLUBS    (“Clubs”),
7     DIAMONDS (“Diamonds”);
8
9     private final String name;
10
11    private Suit(String name)
12      { this.name = name;
13    }
14
15    public String getName()
16      { return name;
17    }
18  }
```


Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Static Imports

- A *static import* imports the static members from a class or Interface:

```
import static <pkg_list>.<class_name>.<member_name>;
OR
import static <pkg_list>.<class_name>.*;
```

- A static import imports members individually or collectively:
- *Use this feature sparingly.*

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU

Summary

- An Array Is An *Object*
- Creating Arrays
- Initializing Arrays
- Command-Line Parameters
- Multidimensional Arrays
- Comparing, Sorting, Searching Arrays
- Enumeration

Copyright 2007-2020, Chen Xudong, School of Software Engineering of BJTU