Java Full Study

Day2: Sep.25

1. import class

   syntax:

```
import java.util.(classname);
```

2. operands vs. operators

   - operands= values, variables, numbers, quantity
   - operators = + - * / %

3. GUI

```
import javax.swing.JOptionPane; // have to import this package

public class Main{
    public static void main(String[] args){
        String name = JOptionPane. showInputDialog("Enter your name");
        JOptionPane.showMessageDialog(null, " Hello" + name);

        int age= Integer.parseInt(JOptionPane.showInputDialog("Enter your
age"));// JOption only accept String, so we should convert to int

        JOptionPane.showMessageDialog(null, "Your age is"+age "years old");
        }
}
```

4. Array

- Define: Datatype[] name={}

```
Example:

String[] cars={"奥迪",  "宝马",  "奔驰"}

car[0]="benchi"

system.out.println(cars[0]);  // print out benchi
```

- 注意array里必须用相同类型的element，比如例子中的array的datatype是string，这时候array里就不能存储integer

```
// 另一种存储方法:
    String[] cars = new String[3];

    for(int i =0; i<cars.length;i++){
    system.out.println(cars[i]);
    }
```

5. 2D Arrays

```
// 如何loop一个2D-Array

String[][] cars = new String[3][3];

cars [0][0]= A;
cars[0][1]=B;
.....
cars[2][2]="Tesla"

for(int i=0; i< cars.length;i++){
    System.out.println();
    for (int j<0;j<cars[i].length;j++){
        System.out.println(cars[i][j]);
}
}
```

6. String method

```
// 一些重要的string的方法

String name = "Bro";

1. 比较两个string是否相同:
boolean result = name.equals("Bro");       //  display: true

2. 比较两个string是否相同, ignore大小写:
boolean result = name.equalsIgnoreCase("bro")    // display: true

3. 得到string的长度:
int result = name.length();  // display: 3

4. 在某个index上的element, 这里是在index=0的时候:
char result =name.charAt(0);  // display: B

5. 判断string是否为空:
boolean result = name.isEmpty();  // display: false

6. 得到某个element的index:
int result = name.indexOf("o");  // display: 2

7. 将string全部换成uppercase, 或全部换成lowercase
String result = name.toUpperCase(); // display: BRO
String result = name.tolowerCase(); // display:bro
```

```
8. 将string中所有的empty space去掉:
String result = name.trim();

9. 将old character替换成 new character:
String result= name.replace('o','a'); // display:BRa
```

7. wrapper classes

```
public class Main {

    public static void main(String[] args) {

        // wrapper class =  provides a way to use primitive data types as
reference data types
        //                  reference data types contain useful methods
        //                  can be used with collections (ex.ArrayList)

        //primitive     //wrapper
        //---------             //-------
        // boolean      Boolean
        // char         Character
        // int          Integer
        // double       Double

        // autoboxing = the automatic conversion that the Java compiler makes
between the primitive types and their corresponding object wrapper classes
        // unboxing = the reverse of autoboxing. Automatic conversion of wrapper
class to primitive

        Boolean a = true;
        Character b = '@';
        Integer c = 123;
        Double d = 3.14;

    }

}
```

8. ArrayList

```
import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {

        // a type of collection
        //ArrayList =   a resizable array.
```

```
        //              Elements can be added and removed after compilation
phase
        //              only store reference data types

        ArrayList<String> food = new ArrayList<String>();

        // Add string to the ArrayList
        food.add("pizza");
        food.add("hamburger");
        food.add("hotdog");

        // 无关例子，只是一些重要method
        food.set(0, "sushi"); // set certain index with string
        food.remove(2); // remove the element on certain index
        food.clear(); // clear the whole arraylist

        for(int i=0; i<food.size(); i++)// With arraylist, usd food.size
instead {
            System.out.println(food.get(i));// Pizza, hamburger,hotdog.
        }
    }
}
```

9. 2D- ArrayList

```
import java.util.*;

public class Main {

    public static void main(String[] args) {

        ArrayList<ArrayList<String>> groceryList = new ArrayList();

        ArrayList<String> bakeryList = new ArrayList();
        bakeryList.add("pasta");
        bakeryList.add("garlic bread");
        bakeryList.add("donuts");

        ArrayList<String> produceList = new ArrayList();
        produceList.add("tomatoes");
        produceList.add("zucchini");
        produceList.add("peppers");

        ArrayList<String> drinksList = new ArrayList();
        drinksList.add("soda");
        drinksList.add("coffee");

        groceryList.add(bakeryList);
        groceryList.add(produceList);
        groceryList.add(drinksList);

        System.out.println(groceryList); // print out three list
        System.out.println(groceryList.get(0)) // print out the first list
        System.out.pringln(groceryList.get(0).get(1)) //print out the second
element in first list, that will be garlic bread
```

```
    }
}
```

10. For-each loop

```
import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {

        // for-each =   traversing technique to iterate through the elements
in an array/collection
        //              less steps, more readable
        //              less flexible


        //String[] animals = {"cat","dog","rat","bird"};
        ArrayList<String> animals = new ArrayList<String>();

        animals.add("cat");
        animals.add("dog");
        animals.add("rat");
        animals.add("bird");


        for(String i : animals) {
            System.out.println(i);
        }
```