W2D1 Homework

1 What’s character of these Collection, List, Set and Map? ArrayList and LinkedList? HashMap and TreeMap?

Answer:

**Collection, List, Set Map**  
 **COLLECTIONS** represents a group of objects, known as its element. Objects can be stored, retrieved and manipulated as elements of collection.  
 **COLLECTION** is a Java Interface  
 **LIST** extends Collection and an instance of List, stores an ordered collection of elements  
 **SET** extends Collection to handle set, must contain unique elements  
 **Map** is an interface, an unordered collection that associates a collection of element values with a set of keys so that elements can be found very quickly.

**ArrayList vs LinkedList  
 ARRAYLIST** is a resizable-array implementation of the List interface. It is used to store an ordered group of elements where duplicates are allowed.  
 **LINKED LIST** doubly-linked list implementation of the List and Deque interfaces. Implements all optional list operations, and permits all elements

**HashMap vs TreeMap  
 HASHMAP** uses a hashtable to implement Map interface. This allows the execution time of basic operations such as get(), and put() to remain constant even for large scale data.  
 **TREE MAP** implements the Map interface thru the use of a Tree. A TreeMap provides an efficient way of storing key/value pairs in sorted order and allows rapid retrieval. Moreover, unlike hashmap, a TreeMap guarantees that elements will be sorted in an ascending key order.

2. （List）Read the codes

**import java.util.\*;**

**public class Test {**

**public static void main(String args[]) {**

**List list = new ArrayList();**

**list.add("Hello");**

**list.add("World");**

**list.add(1, "Learn");**

**list.add(1, "Java");**

**printList(list);**

**}**

**public static void printList(List list) {**

**// 1**

**for (int i = 0; i < list.size(); i++) {**

**System.out.println(list.get(i));**

**}**

**for (Object o : list) {**

**System.out.println(o);**

**}**

**Iterator itor = list.iterator();**

**while (itor.hasNext()) {**

**System.out.println(itor.next());**

**}**

**}**

**}**

Requirement:

1. Write the output of the code.  
   Hello  
   Java  
   Learn  
   World  
   Hello  
   Java  
   Learn  
   World  
   Hello  
   Java  
   Learn  
   World
2. Where and how to modify if change Arraylist with LinkedList?

List list = new ArrayList(); 🡪 List list = new LinkedList();

1. Where and how to modify if change Arraylist with Vector? What’s the difference between ArrayList and Vector?

List list = new ArrayList(); 🡪 List list = new Vector();

**ArrayList** is NOT synchronized while **Vector** is synchronized

3. （List）Write the output of the program.

**import** java.util.\*;

**public** **class** TestList {

**public** **static** **void** main(String args[]) {

List list = **new** ArrayList();

list.add("Hello");

list.add("World");

list.add("Hello");

list.add("Learn");

list.remove("Hello");

list.remove(0);

**for**(**int** i = 0; i < list.size(); i++) {

System.***out***.println(list.get(i));

}

}

}

Output: Hello

Learn

4. Select the right one?

**import** java.util.\*;

**public** **class** TestListSet {

**public** **static** **void** main(String args[]) {

List list = **new** ArrayList();

list.add("Hello");

list.add("Learn");

list.add("Hello");

list.add("Welcome");

Set set = **new** HashSet();

set.addAll(list);

System.***out***.println(set.size());

}

}

1. Compile with error
2. Compile correctly, but throw exception when running.
3. Compile and run well, and output 3 (Answer)
4. Compile and run well, and output 4

5 (List, Map)

**public** **class** Worker {

**private** **int** age;

**private** String name;

**private** **double** salary;

**public** Worker() {

}

**public** Worker (String name, **int** age, **double** salary) {

**this**.name = name;

**this**.age = age;

**this**.salary = salary;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getSalary() {

**return** salary;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

}

**public** **void** work() {

System.***out***.println(name + "is working");

}

}

Please finish the requirement:

1. To create a List and add three workers, and their information shown like this:

|  |  |  |
| --- | --- | --- |
| Name | Age | Salary |
| Simon | 20 | 10000 |
| Jame | 25 | 13000 |
| Alex | 22 | 12000 |

1. Add one worker before Jame ( Steven, 24, 15000)
2. Remove the worker Alex’s information
3. Go through the list using for statement and print out all the worker’s information.
4. Go through the list using Iterator statement to call all the worker’s method work.
5. Over write the equals method for the class Worker. New equals method return true only if the workers’ name, age and salary are the same at the same time.
6. **Sort the all the workers from high to low by salary** and print out the all the workers information with the format “Name: “ + name + “ Salary: “ + salary.
7. Add a id to Worker class, and save the above data to workMap. Map<String, Worker > ( Worker ID, Worker) . **At least three ways t**o go through the workMap, to print out all the workder’s information with Worker id and all other information like “Worker Id: “ + “Name: “ + name + “Age: “ + age + “ Salary: “ + salary.