**Guided Learning Plan for ArrayList**

**Class\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lesson 2 ——ArrayList Method**

|  |  |  |
| --- | --- | --- |
| **Topic** | **details** | **Essential Knowledge** |
| ArrayList Method | * The following are the ArrayList methods that you need to know for the AP CS A exam. * These are included on the AP CS A Java Quick Reference Sheet that you will receive during the exam so you do not need to memorize them. * The E in the method headers below stands for the type of the element in the ArrayList; this type E can be any Object type. We will look at how these methods work below.  |  |  | | --- | --- | | * **int size()** | returns the number of elements in the list | | * **boolean add(E obj)** | appends obj to the end of the list and returns true | | * **void add(int index, E obj)** | moves any current objects at index or beyond to the right (to a higher index) and inserts obj at the index | | * **E remove(int index)** | removes the item at the index and shifts remaining items to the left (to a lower index) | | * **E set(int index, E obj)** | replaces the item at index with obj | | * **E get(int index)** | returns the item in the list at the index | | |
| size( ) | You can get the number of items in a ArrayList using its size( ) method. The ArrayList starts out empty with a size of 0.  Syntax:  ArrayList<String> list = new ArrayList<String>();  System.out.println( list.size() );//result is 0 | |
| add(E obj) | * You can add values to an ArrayList by using the method add(obj) which will add the object to the **end of the list**, just like you would join the end of the line to board a bus. * Notice that we added the same string to the list more than once. Lists can hold duplicate objects.      * Attention! To add objects to an ArrayList, the objects must be of the **SAME** data type to instantiate the ArrayList!   QQuiz:  // add an integer 5 to the following Arraylist  ArrayList<Integer> list = new ArrayList<Integer>();  Write your answer here:  Java provide autoboxing & unboxing function, so you can just add the int value directly like add(5) in any Java version and it will be changed into an Integer object automatically. This is called autoboxing. When you pull an int value out of a list of Integers that is called unboxing.   * You can put any kind of Objects into an ArrayList. Even objects for a class that you wrote. For example, here is an ArrayList of Students. See StudentList.java | |
| add(int dex, E obj) | There are actually two different add methods in the ArrayList class. The add(obj) method adds the passed object to the end of the list.  The add(index,obj) method adds the passed object at the passed index, but first moves over any existing values to higher indicies to make room for the new object.  Quiz:  What will print when the following code executes?  ArrayList<Integer> list1 = new ArrayList<Integer>();  list1.add(1);  list1.add(2);  list1.add(3);  list1.add(2, 4);  list1.add(5);  System.out.println(list1);  A. [1, 2, 3, 4, 5] B. [1, 4, 2, 3, 5] C. [1, 2, 4, 3, 5] D. [1, 2, 4, 5] | |
| remove(int index) | You can also remove values from an ArrayList by using **remove(index)** to remove the item at the given index from the list. This will move all the other items over in the underlying array and decrease the size of the ArrayList by 1.  The remove(int index) method will **remove the object at the index** and shift left any values to the right of the current index. It doesn’t remove the object that matches the integer value given.    Quiz:  What will print when the following code executes?  List<Integer> list1 = new ArrayList<Integer>();  list1.add(1);  list1.add(2);  list1.add(3);  list1.remove(2);  System.out.println(list1);  A. [2, 3] B. [1, 2, 3] C. [1, 2] D. [1, 3] | |
| ArrayList get/set Methods | * You can get the object at an index using   obj = listName.get(index)   * and set the object at an index using   listName.set(index,obj)  Set/Get are used after you add and remove elements to an ArrayList to change or retrieve them. | Notice that ArrayLists use set/get methods instead of using the square brackets array[index] that arrays use. This is because ArrayList is a class with methods that provide access to the underlying array. |
| ArrayList<String> shoppingList = new ArrayList<String>()  shoppingList .add(“carrots”);  shoppingList.add("bread");  System.out.println(shoppingList);  //result si： [carrots, bread] | |
| Comparing arrays and ArrayLists | When do you use arrays and when do you use ArrayLists?   * Use an array when you want to store several items of the same type and you know how many items will be in the array and the items in the array won’t change in order or number. * Use an ArrayList when you want to store several items of the same type and you don’t know how many items you will need in the list or when you want to remove items from the list or add items to the list while the program is running. * Here is a comparison of how to create arrays and ArrayLists:   // arrays must specify a size!  int[ ] highScores = new int[5];  String[ ] names = new String[5];  // ArrayLists are empty to start with  ArrayList<Integer> highScoreList = new ArrayList<Integer>();  ArrayList<String> nameList = new ArrayList<String>();   * Here is a comparison of how to access and change elements in arrays and ArrayLists      | Operation | array | ArrayList | | --- | --- | --- | | length/size | array.length | list.size() | | Access | value = array[index]; | value = list.get(index); | | Modify | array[index] = value; | list.set(index,value); | |  |  |  |  * Note that the ArrayList methods add and remove **do not have** a simple equivalent in arrays because they actually change the size of the underlying array and move elements over. | |
| Programming Challenge | * Write an ArrayList toDoList, and store “Do homework”,”Help make dinner”and “Call grandma” [in](C:/Users/zhang/AppData/Local/youdao/dict/Application/8.9.9.0/resultui/html/index.html" \l "/javascript:;) [sequence](C:/Users/zhang/AppData/Local/youdao/dict/Application/8.9.9.0/resultui/html/index.html" \l "/javascript:;). * Add “Watch TV”to the toDoList before ”call grandma” * Change “Help make dinner” in toDoList to “Order pizza” * Remove “Do homework” and move everything down * Print the finally toDoList | |
| Summary  (ESSENTIAL KNOWLEDGE  ) | 1. The ArrayList class is part of the java.util package. An import statement can be used to make this class available for use in the program. 2. The following ArrayList methods—including what they do and when they are used—are part of the Java Quick Reference | |

**Lesson 2 ——ArrayList Method**

Quiz

1、Consider the following code segment.

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

animals.add("fox");

animals.add(0, "squirrel");

animals.add("deer");

animals.set(2, "groundhog");

animals.add(1, "mouse");

System.out.println(animals.get(2) + " and " + animals.get(3));

What is printed as a result of executing the code segment?

A mouse and fox

B fox and groundhog

C groundhog and deer

D fox and deer

E squirrel and groundhog

2、Consider the following code segment.

ArrayList<Integer> oldList = new ArrayList();

oldList.add(100);

oldList.add(200);

oldList.add(300);

oldList.add(400);

ArrayList<Integer> newList = new ArrayList();

newList.add(oldList.remove(1));

newList.add(oldList.get(2));

System.out.println(newList);

What, if anything, is printed as a result of executing the code segment?

A [100, 300, 400]

B [200, 300]

C [200, 400]

D Nothing is printed because the code segment does not compile.

E Nothing is printed because an IndexOutOfBoundsException will occur.