Programming and Data Structures
Active Learning Activity 5: Generics

Activity Objectives

At the end of this activity, students should be able to:

1. Create a generic class with two generic types.
2. Use the generic class ***ArrayList<E> (java.util.ArrayList)*** and the generic interface ***Comparator<E>*** (***java.util.Comparator***).
3. Create a generic method to search in an ***ArrayList***.
4. Write a Java program that uses generic classes, interfaces, and methods.

Activity

---

1. Create the generic class ***Pair<E1, E2>*** with two generic types as seen in class. The class should have appropriate constructors, accessors, mutators, ***toString()***, and ***equals()*** methods as seen in class.

2. Create the class ***ComparatorByFirst*** that implements the interface ***Comparator*** for the type ***Pair<E1, E2>***. Define the method ***compare*** to compare the first elements of two pairs.

3. Create the class ***ComparatorBySecond*** that implements the interface ***Comparator*** for the type ***Pair<E1, E2>***. Define the method ***compare*** to compare the second elements of two pairs.

4. Create the test class ***Test***. In the main method, create an instance of the generic class ***ArrayList*** for the type ***Pair<String, String>*** and name it ***states***. Each element in ***states*** contains the name of a state and its capital. Read the list of states and their capitals from the file `states.txt` and store the information in the array list ***states***.

5. Create another instance of the generic class ***ArrayList*** for the type ***Pair<String, Integer>*** and name it ***trees***. Each element in ***trees*** contains the name of a tree and its maximum height. Read the list of trees and their heights from the file `trees.txt` and store the information in the array list ***trees***.

6. Define a generic method ***search*** in class ***Test*** with the header below.

```
public static <E1, E2> int search(
                        ArrayList<Pair<E1, E2>> list, E1 key)
```

search returns the index where ***key*** is found in ***list*** or -1 otherwise. Note that ***key*** is compared to the first element of the pair (type ***E1***).

7. Prompt the user to select one of the following operations:

   a. ***View the list of states***: display the content of the array list ***states***.

   b. ***Search for a state capital***: prompt the user to enter the name of a state and call ***search()*** to look for it in the list ***states***. Display the state and its capital if the state is found or the message "`State not found`".

   c. ***Sort states by name***: sort the array list ***states*** using the method ***sort*** from the class ***ArrayList***. Pass an anonymous object of type ***ComparatorByFirs***t as the argument to ***sort***. Display the sorted list of states.

   d. ***Sort states by capital***: sort the array list ***states*** using the method ***sort*** from the class ***ArrayList***. Pass an anonymous object of type ***ComparatorBySecond*** as the argument to ***sort***. Display the sorted list of states.

   e. ***View the list of trees***: display the content of the array list ***trees***.

   f. ***Search for a tree***: prompt the user to enter the name of a tree and call ***search()*** to look for it in the list ***trees***. Display the tree and its height if the tree is found or the message "`Tree not found`".

   g. ***Sort trees by name***: sort the array list ***trees*** using the method ***sort*** from the class ***ArrayList***. Pass an anonymous object of type ***ComparatorByFirs***t as the argument to ***sort***. Display the sorted list of trees.

h. ***Sort trees by height***: sort the array list ***trees*** using the method ***sort*** from the class ***ArrayList***. Pass an anonymous object of type ***ComparatorBySecond*** as the argument to ***sort***. Display the sorted list of trees.

i. ***Exit***: exit the program

8. Submit the files ***Pair.java***, ***ComparatorByFirst.java***, ***ComparatorBySecond.java,*** and ***Test.java*** on Github.

========== Sample output ==========

```
Select an operation:
1: View list of states
2: Search for a state
3: Sort states by name
4: Sort states by capital
5: View list of trees
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
1
(Alabama, Montgomery)
(Alaska, Juneau)
(Arizona, Phoenix)
(Arkansas, Little Rock)
(California, Sacramento)
(Colorado, Denver)
(Connecticut, Hartford)
(Delaware, Dover)
(Florida, Tallahassee)
(Georgia, Atlanta)
(Hawaii, Honolulu)
(Idaho, Boise)
(Illinois, Springfield)
(Maryland, Annapolis)
(Minnesota, Saint Paul)
(Iowa, Des Moines)
(Maine, Augusta)
(Kentucky, Frankfort)
(Indiana, Indianapolis)
(Kansas, Topeka)
(Louisiana, Baton Rouge)
(Oregon, Salem)
(Oklahoma, Oklahoma City)
(Ohio, Columbus)
```

```
(North Dakota, Bismarck)
(North Carolina, Raleigh)
(New York, Albany)
(New Mexico, Santa Fe)
(New Jersey, Trenton)
(New Hampshire, Concord)
(Nevada, Carson City)
(Nebraska, Lincoln)
(Montana, Helena)
(Missouri, Jefferson City)
(Mississippi, Jackson)
(Massachusetts, Boston)
(Michigan, Lansing)
(Pennsylvania, Harrisburg)
(Rhode Island, Providence)
(South Carolina, Columbia)
(South Dakota, Pierre)
(Tennessee, Nashville)
(Texas, Austin)
(Utah, Salt Lake City)
(Vermont, Montpelier)
(Virginia, Richmond)
(Washington, Olympia)
(West Virginia, Charleston)
(Wisconsin, Madison)
(Wyoming, Cheyenne)

Select an operation:
1: View list of states
2: Search for a state
3: Sort states by name
4: Sort states by capital
5: View list of trees
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
5
(White cedar, 50)
(Black Ash, 70)
(White Ash, 80)
(Bigtooth Aspen, 80)
(Basswood, 130)
(American Beech, 70)
(Black Birch, 50)
(Butternut, 60)
(Black Cherry, 80)
(American Chestnut, 90)
(Cottonwood, 100)
(American Elm, 80)
```

```
(Hawthorn, 30)
(Hemlock, 100)
(Shagbark Hickory, 80)
(American Hornbeam, 40)
(American Larch, 60)
(Black Locust, 82)
(Red Maple, 90)
(Sugar Maple, 75)
(Black Oak, 150)
(White Oak, 80)
(Red Pine, 75)
(White Pine, 210)
(Sycamore, 100)
(Tulip Tree, 90)
(Black Walnut, 75)
(Black Willow, 100)

Select an operation:
1: View list of states
2: Search for a state
3: Sort states by name
4: Sort states by capital
5: View list of trees
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
2
Enter a state name:
Pennsylvania
State found: (Pennsylvania, Harrisburg)

Select an operation:
1: View list of states
2: Search for a state
3: Sort states by name
4: Sort states by capital
5: View list of trees
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
2
Enter a state name:
Misouri
State not found.

Select an operation:
1: View list of states
2: Search for a state
```

```
3: Sort states by name
4: Sort states by capital
5: View list of trees
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
6
Enter a tree name:
Red Maple
Tree found: (Red Maple, 90)

Select an operation:
1: View list of states
2: Search for a state
3: Sort states by name
4: Sort states by capital
5: View list of trees
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
6
Enter a tree name:
Key Lime
Tree not found.

Select an operation:
1: View list of states
2: Search for a state
3: Sort states by name
4: Sort states by capital
5: View list of trees
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
3

(Alabama, Montgomery)
(Alaska, Juneau)
(Arizona, Phoenix)
(Arkansas, Little Rock)
(California, Sacramento)
(Colorado, Denver)
(Connecticut, Hartford)
(Delaware, Dover)
(Florida, Tallahassee)
(Georgia, Atlanta)
(Hawaii, Honolulu)
```

```
(Idaho, Boise)
(Illinois, Springfield)
(Indiana, Indianapolis)
(Iowa, Des Moines)
(Kansas, Topeka)
(Kentucky, Frankfort)
(Louisiana, Baton Rouge)
(Maine, Augusta)
(Maryland, Annapolis)
(Massachusetts, Boston)
(Michigan, Lansing)
(Minnesota, Saint Paul)
(Mississippi, Jackson)
(Missouri, Jefferson City)
(Montana, Helena)
(Nebraska, Lincoln)
(Nevada, Carson City)
(New Hampshire, Concord)
(New Jersey, Trenton)
(New Mexico, Santa Fe)
(New York, Albany)
(North Carolina, Raleigh)
(North Dakota, Bismarck)
(Ohio, Columbus)
(Oklahoma, Oklahoma City)
(Oregon, Salem)
(Pennsylvania, Harrisburg)
(Rhode Island, Providence)
(South Carolina, Columbia)
(South Dakota, Pierre)
(Tennessee, Nashville)
(Texas, Austin)
(Utah, Salt Lake City)
(Vermont, Montpelier)
(Virginia, Richmond)
(Washington, Olympia)
(West Virginia, Charleston)
(Wisconsin, Madison)
(Wyoming, Cheyenne)

Select an operation:
1: View list of states
2: Search for a state
3: Sort states by name
4: Sort states by capital
5: View list of trees
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
```

4

```
(New York, Albany)
(Maryland, Annapolis)
(Georgia, Atlanta)
(Maine, Augusta)
(Texas, Austin)
(Louisiana, Baton Rouge)
(North Dakota, Bismarck)
(Idaho, Boise)
(Massachusetts, Boston)
(Nevada, Carson City)
(West Virginia, Charleston)
(Wyoming, Cheyenne)
(South Carolina, Columbia)
(Ohio, Columbus)
(New Hampshire, Concord)
(Colorado, Denver)
(Iowa, Des Moines)
(Delaware, Dover)
(Kentucky, Frankfort)
(Pennsylvania, Harrisburg)
(Connecticut, Hartford)
(Montana, Helena)
(Hawaii, Honolulu)
(Indiana, Indianapolis)
(Mississippi, Jackson)
(Missouri, Jefferson City)
(Alaska, Juneau)
(Michigan, Lansing)
(Nebraska, Lincoln)
(Arkansas, Little Rock)
(Wisconsin, Madison)
(Alabama, Montgomery)
(Vermont, Montpelier)
(Tennessee, Nashville)
(Oklahoma, Oklahoma City)
(Washington, Olympia)
(Arizona, Phoenix)
(South Dakota, Pierre)
(Rhode Island, Providence)
(North Carolina, Raleigh)
(Virginia, Richmond)
(California, Sacramento)
(Minnesota, Saint Paul)
(Oregon, Salem)
(Utah, Salt Lake City)
(New Mexico, Santa Fe)
(Illinois, Springfield)
(Florida, Tallahassee)
```

```
(Kansas, Topeka)
(New Jersey, Trenton)

Select an operation:
1: View list of states
2: Search for a state
3: Sort states by name
4: Sort states by capital
5: View list of trees
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
7

(American Beech, 70)
(American Chestnut, 90)
(American Elm, 80)
(American Hornbeam, 40)
(American Larch, 60)
(Basswood, 130)
(Bigtooth Aspen, 80)
(Black Ash, 70)
(Black Birch, 50)
(Black Cherry, 80)
(Black Locust, 82)
(Black Oak, 150)
(Black Walnut, 75)
(Black Willow, 100)
(Butternut, 60)
(Cottonwood, 100)
(Hawthorn, 30)
(Hemlock, 100)
(Red Maple, 90)
(Red Pine, 75)
(Shagbark Hickory, 80)
(Sugar Maple, 75)
(Sycamore, 100)
(Tulip Tree, 90)
(White Ash, 80)
(White Oak, 80)
(White Pine, 210)
(White cedar, 50)

Select an operation:
1: View list of states
2: Search for a state
3: Sort states by name
4: Sort states by capital
5: View list of trees
```

```
6: Search for a tree
7: Sort trees by name
8: Sort trees by height
9: Exit
8

(Hawthorn, 30)
(American Hornbeam, 40)
(Black Birch, 50)
(White cedar, 50)
(American Larch, 60)
(Butternut, 60)
(American Beech, 70)
(Black Ash, 70)
(Black Walnut, 75)
(Red Pine, 75)
(Sugar Maple, 75)
(American Elm, 80)
(Bigtooth Aspen, 80)
(Black Cherry, 80)
(Shagbark Hickory, 80)
(White Ash, 80)
(White Oak, 80)
(Black Locust, 82)
(American Chestnut, 90)
(Red Maple, 90)
(Tulip Tree, 90)
(Black Willow, 100)
(Cottonwood, 100)
(Hemlock, 100)
(Sycamore, 100)
(Basswood, 130)
(Black Oak, 150)
(White Pine, 210)

Select an operation:
1: View list of states
2: View list of trees
3: Search for a state
4: Search for a tree
5: Sort states by name
6: Sort states by capital
7: Sort trees by name
8: Sort trees by height
9: Exit
9
```