Name: Lab11\_Knowlton

Author: Matthew Knowlton

Lab Number: 11

Date: 25 April 2022

BEGIN main

1. DECLARE listOfCities array of size 20 to hold City objects.
2. DECLARE each element of listOfCities as a new object using City class constructor and name, state, size, population, and elevation as parameters.
3. DECLARE scanner as a new object using Scanner class constructor and System.in as a parameter to get user input.
4. PRINT “Welcome to the List of Cities!”
5. DO WHILE true
   1. PRINT selection menu
   2. PROMPT user for an integer from the menu and assign it to userSelection
   3. IF userSelection is equal to 2 or 3
      1. PRINT corresponding question for further user input
   4. SWITCH based on userSelection
      1. CASE userSelection is 1
         1. CALL print function with listOfCities as a parameter to print the table of the list of cities
      2. CASE userSelection is 2
         1. CALL searchByCity function using listOfCities and scanner input as parameters to print a table of cities matching the scanner input
      3. CASE userSelection is 3
         1. CALL searchByState function using listOfCities and scanner input as parameters to print a table of cities from the state that matches the scanner input.
      4. CASE userSelection is 4
         1. CALL maxPopulation using listOfCities as a parameter to get a table with the city whose population is the largest.
      5. CASE userSelection is 5
         1. CALL highestElevation using listOfCities as a parameter to get a table with the city whose elevation is the highest.
      6. CASE userSelection is 6
         1. CALL exit function using scanner as a parameter to close the scanner, print a goodbye message, and exit the program.
   5. END SWITCH
6. END DO

END

BEGIN print(City[] listOfCities)

1. CALL printHeader to print the header of the table
2. FOR each city element from listOfCities
   1. CALL printCity using city as a parameter to print the current city
3. END FOR

END

BEGIN searchByCity(City[] listOfCities, String name)

1. DECLARE a boolean named found as false
2. FOR each city element from listOfCtiies
   1. IF the result of the function city.getName is equal to name
      1. IF the boolean found is equal to false
         1. CALL printHeader to print the header of the table
      2. Call printCity using city as the parameter to print the city information
      3. SET the found boolean to true
   2. END IF
3. END FOR

END

BEGIN searchByState(City[] listOfCities, String state)

1. DECLARE a boolean named found as false
2. FOR each city element from listOfCtiies
   1. IF the result of the function city.getState is equal to state
      1. IF the boolean found is equal to false
         1. CALL printHeader to print the header of the table
      2. Call printCity using city as the parameter to print the city information
      3. SET the found boolean to true
   2. END IF
3. END FOR

END

BEGIN maxPopulation(City[] listOfCities)

1. DECLARE a string named maxCity using the result of the first element from listOfCities and calling its function getName
2. DECLARE a string named maxPopulation using the result of the frist element from listOfCities and calling its function getPopulation
3. FOeah city element from listOfCities
   1. SET maxCity using a ternary operation
      1. IF the result of city.getPopulation is larger than the current value of maxPopulation
         1. SET maxCity equal to the result of city.getName
      2. ELSE
         1. SET maxCity equal to the current value of maxCity
   2. SET maxPopulation using a ternary operation
      1. IF the result of city.getPopulation is larger than the current value of maxPopulation
         1. SET maxPopulation equal to the result of city.getPopulation
      2. ELSE
         1. SET maxPopulation equal to the current value of maxPopulation
4. END FOR
5. CALL searchByCity using listOfCities and maxCity as parameters to print the city information of the city with the largest population.

END

BEGIN highestElevation(City[[] listOfCities)

1. DECLARE a string named maxCity using the result of the first element from listOfCities and calling its function getName
2. DECLARE a string named maxElevation using the result of the first element from listOfCities and calling its function getElevation
3. FOR each city element from listOfCities
   1. SET maxCity using a ternary operation
      1. IF the result of city. getElevation is larger than the current value of maxPopulation
         1. SET maxCity equal to the result of city.getName
      2. ELSE
         1. SET maxCity equal to the current value of maxCity
   2. SET maxElevation using a ternary operation
      1. IF the result of city.getPopulation is larger than the current value of maxPopulation
         1. SET maxElevation equal to the result of city.getElevation
      2. ELSE
         1. SET maxElevation equal to the current value of maxElevation
4. END FOR
5. CALL searchByCity using listOfCities and maxCity as parameters to print the city information of the city with the highest elevation.

END

BEGIN printHeader()

1. PRINT a formatted string using the fields of name, state, size, population, and elevation.

END

BEGIN printCity(City city)

1. PRINT a formatted string using the fields of city.getName, city.getState, city.getSize, city.getPopulation, and city.getElevation

END

BEGIN exit(Scanner scanner)

1. CALL the close function from the scanner object
2. PRINT “Goodbye!”
3. CALL System.exit using 0 as the parameter.

END