CS101- Algorithms and Programming I

Lab 06

Lab Objectives: ArrayLists.

IMPORTANT:

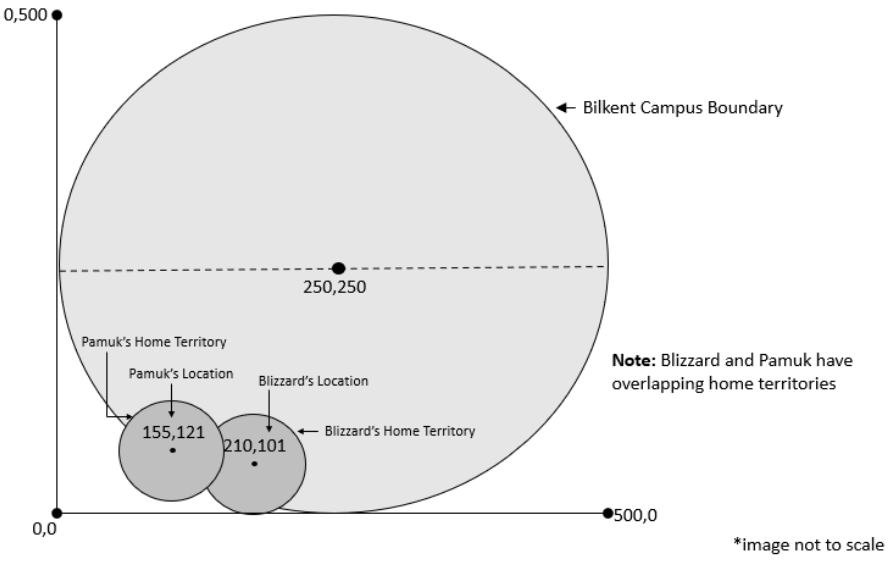
- For all labs in CS 101, your solutions must conform to the CS101 style guidelines (rules!).
- You should not use break statements in your solution.
- Before implementing your programs, you should analyse the problem and plan and write the algorithm you will use to solve the problem.

1. The animal lovers at Bilkent University have noticed that when the number of cats in a given campus location increases, the likelihood that the cats will be injured (fighting, predators, traffic, etc.) also increases. Therefore, they have made the decision that if there are 3 or more cats in overlapping 'home' territories on campus, then the cat(s) who have been at the location for the least amount of time should be relocated.

   Your application will use 3 `ArrayLists`, one to store the names of the cats, one to store the location of each cat (x,y) and the third to store the number of months the cat has been at the current location. The `ArrayLists` are parallel lists, meaning that the cat whose name is at index 0 of the name list, will have their location at the same index in the locations and months lists.

   Assumptions:

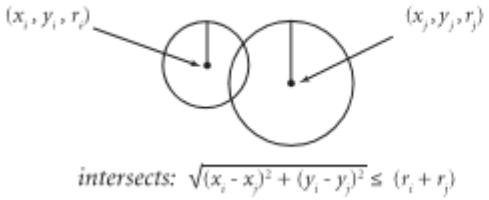
   - Locations will be stored as Strings, "x_coordinate,y_coordinate". Example: "420,80".
   - The home territory of each cat is a circle with a radius of 50 metres, where the x,y coordinate is the centre of the territory.
   - Each cat's location (x,y coordinates) will be within the Bilkent Campus, which has a circular boundary and a diameter of 500m. The centre of campus is at location 250,250.

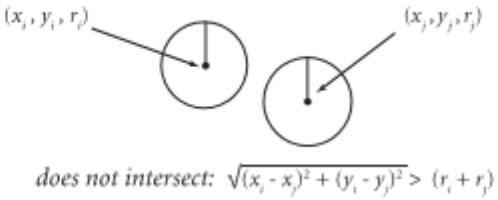


3. Download the file `CatInfoGenerator.java`, and complete the implementation of the `getRandomNameList()` method. The method should add `num` random names from the global static `catNames` list to a new ArrayList and return the random list of names. There should be no duplicates, and you should make sure that `num` is less than the number of names in the `catNames` list, if not, `num` should be set to the size of the `catNames` list.
4. Your application should implement several helper methods. Below are suggested methods, but you may use others if you wish:
    - `double calculateDistance()`: takes 2 String locations ("x,y") as parameters and returns the distance between the two locations. To calculate the distance between two points, use the following formula:

$$d=\sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}$$

    - `boolean doesIntersect()`: takes 2 String locations as parameters and returns true if the circles with the given centre points intersect, false if not. Two circles intersect if the distance between their centres is less than or equal to the sum of their radii.





    - `int countIntersecting()`: takes a location and an ArrayList of locations as parameters and counts and returns the number of locations in the ArrayList that intersect with the given location.
    - `displayIntersections()`: takes a list of cats, locations and months as parameters, and for each cat, displays the information about the cats with intersecting home territories.
    - `findMinMonthsAtLocation()`: takes a String location and an ArrayList of locations and an ArrayList of months as parameters, and finds all cats whose territories intersect with the given String location, and returns the index of the cat who has been in their intersecting location for the minimum number of months.
    - `findRandomLocation()`: returns a String location ("x,y" coordinate) within the bounds of the Bilkent campus.
    b. `moveCats()`: takes 2 lists, locations and months, and find all locations with 3 or more intersecting cats. For each location if there are 3 or more cats whose home territories intersect (represented by a circle with radius 50m and center point with the given x,y location), move the cat who has been at the location for the minimum number of months. The territory of the new location should have no cats with intersecting home territories.

5. Your application should do the following:
    a. Download `CatInfoGenerator` and using the static methods provided in `CatInfoGenerator` (that you will complete the implementation of), create 3 ArrayLists that store the random names, locations, and months for 20 cats.
    b. Display the information about each cat and the cats whose locations they intersect with.
    c. If there are 3 or more cats in intersecting home territories (represented by a circle with radius 50m and centre point with the given x,y location), move the cat(s) who have been at the location for the minimum number of months. The territory of the new location should not already have any cats with intersecting home territories.
    d. Display again the information about each cat and the cats whose locations they intersect with.

Sample Run:

```
Before moving the cats:

Aslan at location 50,307 for 29 months intersects with: NO CATS

Karamel at location 122,366 for 34 months intersects with: NO CATS

Tekir at location 63,121 for 58 months intersects with:
        Splatter, at location 30,125 for 27 months
Splatter at location 30,125 for 27 months intersects with:
        Tekir, at location 63,121 for 58 months
Benek at location 379,270 for 12 months intersects with:
        Blizzard, at location 365,225 for 45 months
        Patches, at location 375,244 for 49 months
Lucky at location 32,201 for 47 months intersects with: NO CATS

Blizzard at location 365,225 for 45 months intersects with:
        Benek, at location 379,270 for 12 months
        Patches, at location 375,244 for 49 months
Patches at location 375,244 for 49 months intersects with:
        Benek, at location 379,270 for 12 months
        Blizzard, at location 365,225 for 45 months
Gora at location 452,66 for 55 months intersects with: NO CATS

Zeytin at location 143,192 for 22 months intersects with: NO CATS

After moving the cats:

Aslan at location 50,307 for 29 months intersects with: NO CATS

Karamel at location 122,366 for 34 months intersects with: NO CATS

Tekir at location 63,121 for 58 months intersects with:
        Splatter, at location 30,125 for 27 months
Splatter at location 30,125 for 27 months intersects with:
        Tekir, at location 63,121 for 58 months
Benek at location 136,60 for 0 months intersects with: NO CATS

Lucky at location 32,201 for 47 months intersects with: NO CATS

Blizzard at location 365,225 for 45 months intersects with:
        Patches, at location 375,244 for 49 months
Patches at location 375,244 for 49 months intersects with:
        Blizzard, at location 365,225 for 45 months
Gora at location 452,66 for 55 months intersects with: NO CATS

Zeytin at location 143,192 for 22 months intersects with: NO CATS
```