MapsApp

Getting Started

To run the project

1. Import the project to NetBeans;
2. Change the *FILENAME* variable inside *MapService.java* to exact location of the reference points;
3. Build and run;

File Structures

- *src*
  - *java* - Folder contains controllers implementation
    - **MapController.java** - main controller, which implements three actions;
    - **MapService.java** - service which initialize RTree during application startup;
    - **Location.java** - model class for reference points in map;
    - **RTree.java** - RTree implementation.
  - web
    - *WEB-INF*: Folder contains jsp files;
      - **map.jsp**: jsp file, which displays the map to client;

Navigation

1. **url/map** - returns the map.jsp page, which displays the map.
2. **url/nearby?x={float}&y={float}** - returns the 10 closest points to the location(x,y), where x - latitude and y - longitude, in JSON format.
3. **url/area?x1={float}&y1={float}&x2={float}&y2={float}** - returns the points bounded by (x1,y1),(x2,y2) in JSON format.

Class Documentations

MapController.java

Functions

- **map** - action, returns map.jsp;
- **nearby** - action, returns near by reference points (Location class objects) according to passed parameters;
- **area** - action, returns reference points (Location objects) inside specified rectangle. Rectangle is constructed by top-left and bottom-right points passed as parameters to action;

MapService.java

Functions

- **loadTree** - function to load the data from *NationalFile*. To load it from different file you need to change the **filename**;
- **search** - performs the search operation according to *coordinates* and *dimension*. Returns the ArrayList of Location; ######Members
- **proviceHash** - *HashMap<String, Integer>* - used to store the mapping from "state name + provice name" into "unique integer". This is used to save the space. Saving integers instead of saving the state and provice names requires less memory;
- **reverseProviceHash** - *HashMap<Integer, String>* - used to store the mapping from "unique integer" into "state name + provice name". Used to restore the state and provice name from unique integers;

RTree.java

Functions

- **insert** - inserts an entry to the Rtree. If the size of node is more than maximum number of possible entries splits the node.
- **search** - search for an entry in Rtree;
- **adjustTree** - adjusts the tree;
- **tighten** - tightens the node;
- **getArea** - returns the area calculated by multiplying the dimensions;
- **isOverlap** - checks if the passed coordinates and dimensions overlap with each other;
- **splitNode** - splits the node;

Location.java

Functions

- **setDistance** - calculate the distance from its coordinates to given coordinates;
- **setters** and **getters**