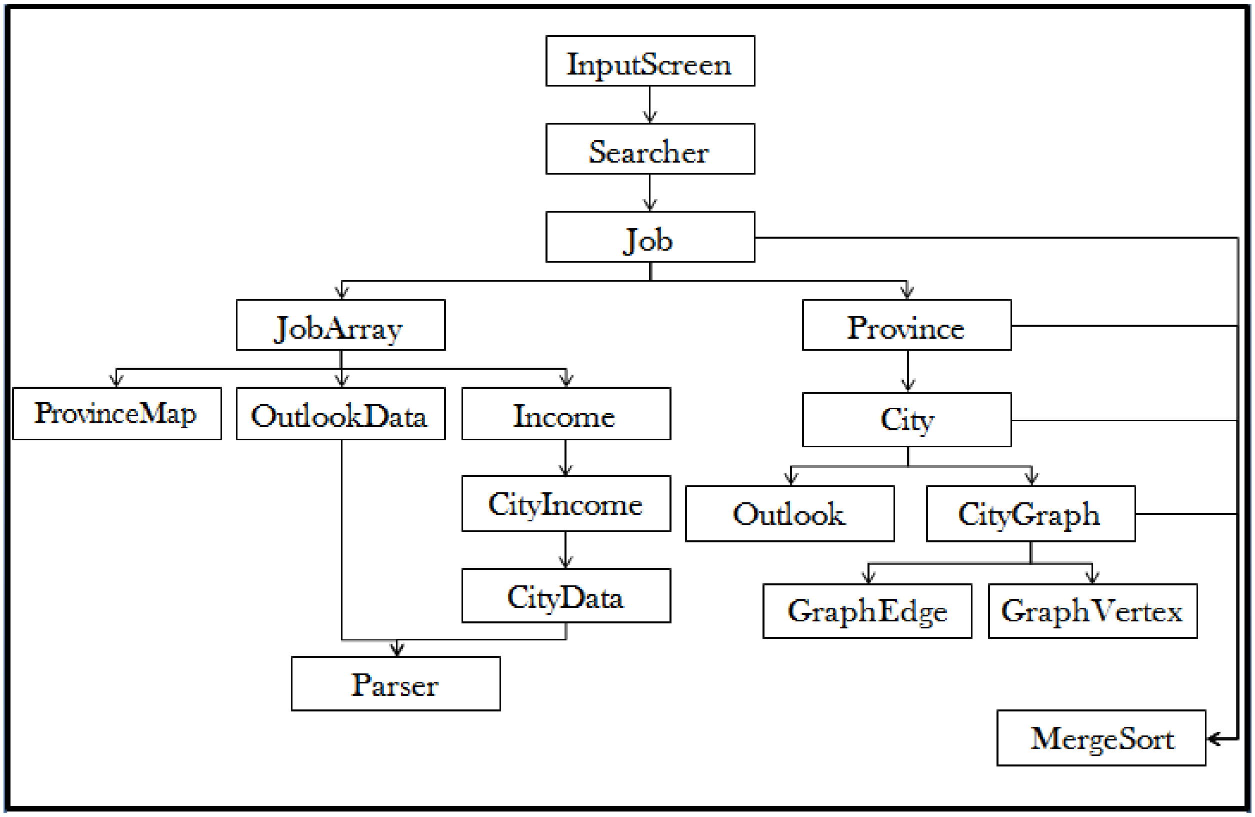
UML Diagram



1: See Expanded Classes Below

1. Input Screen:

InputScreen
frame: Jframe
panel: Jpanel
label: Jlabel
textName: JTextField
textProvince: JTextField
textIncome: JTextField
textArea: JTextArea
button: Jbutton

2. Searcher:

Searcher
jobsFetcher: jobArray
jobs: ArrayList<Job>
searchCity(String jobName)
searchProvinceCity(String jobName, String provinceCode)
searchIncomeCity(String jobName, String provinceCode, String aveIncome)

3. Job:

Job
name: String
provinces: ArrayList<Provinces>
getName()
getProvinces()
addOutlook(Province province, City city, int potential, String trend)
getPotential(Province province, City city)
getAvgPotential(Province province)
getCities()
getCities(Province province)
searchJob(ArrayList<Job> jobs)
sortProvinces()
sortCity()
equals(Object that)
toString()

4. JobArray:

JobArray
jobs: ArrayList<Job>
incomeFetcher: Income
income: ArrayList<CityIncome>
map: ProvinceMap
toJob
getJobArray()

5. Province:

Province
provinceCode: String
avgPotential: double
provinceIncome: double
job: Job
cities: ArrayList<City>
provinceTrends: ArrayList<String>
getProvinceCode()
getJob()
getAvgPotential()
getCities()
getProvinceTrends()
getProvinceName()
getProvinceIncome()
setProvinceIncome(double provinceIncome)
addOutlook(City city, int potential, String trend)
getPotential(City city)
searchProvince(ArrayList<Province> provinces)
sortCities()
compareTo(Object thatP)
toString()

6. ProvinceMap:

ProvinceMap
f: String
forward: Map<String, String>
backward: Map<String, String>
add(String key, String value)
getForward(String key)
getBackward(String key)

7. OutlookData:

OutlookData
Title: String
CPP: String
Trends: String
TrendsDate: String
Lang: String
ProvAbbr: String
Location: String
potential: int
code: int
provID: int
NOC: int
cityID: int
getNOC()
getCode()
getCPP()
getLang()
getLocation()
getPotential()
getProvAbbr()
getProvID()
getTitle()
getTrends()
getTrendsDate()
getCityID()

8. Income:

Income
cities: ArrayList<CityIncome>
toCityIncome(CityData[] c)
searchCity(String cityName)
getCities()

9. City:

City
cityName: String
province: Province
cityOutlook: Outlook
income: double
getCityName()
getProvince()
getOutlook()
setOutlook()
searchCity(ArrayList<City> cities)
compareTo(Object that)
getIncome()
toString()

10. CityIncome:

CityIncome
cityName: String
province: String
incomes: ArrayList<Double>
avgIncome: double
getCityName()
getIncomes()
addIncome(double income)
getAvgIncome()

11. Outlook:

Outlook
potential: int
trend: String
city: City
getPotential()
getTrend()
getCity()
toString()

12. CityGraph:

CityGraph
SCALING_FACTOR: double
CITIES_TO_RETURN: int
crawlDummies(GraphVertex source, double targetValue)
getRelatedCities(City toFind)

13. CitaData:

CityData
year: int
city: String
province: String
GeographicalID: String
DataType: String
Vector: String
Coordinate: double
Income: double
getCoordinate()
getDataType()
getGeographicalID()
getProvince()
getCity()
getIncome()
getVector()
getYear()

14. GraphEdge:

GraphEdge
v: GraphVertex
w: GraphVertex
weight: long
getConnection(GraphVertex c)
getWeight()
compareTo(Object oThat)

15. GraphVertex:

GraphVertex
city: City
outlook: double
adj: ArrayList<GraphEdge>
connect(GraphVertex other, long weight)
getAdj()
getCityName()
getCity()
getOutlook()

16. Parser:

Parser
outlookName: String
incomeName: String
getOutlookDataArray()
getCityDataArray()

17. MergeSort:

MergeSort
aux: ArrayList<Comparable>