**<<Interface>>**
**Comparable<T>**

+compareTo(otherObject: T): int

---

**TeamMate**

#errors: String
# formatter: SimpleDateFormat
-reader: CustomBufferedReader
#results: String

+TeamMate():void
addStringListToCountry(ArrayList<String>, Country): void
convertCSVToStringList(fileName: String): ArrayList<String>
convertStringListToArrayListOfTeams(ArrayList<String>): ArrayList<Team>
convertStringToPerson(personString: String, formatter: SimpleDateFormat): Person
convertStringToTeam(line: String): Team
convertToCSV(String):void
main(args: String[]): void
promptUser(country:Country)

---

**Country**

-states: ArrayList<State>
treeMapNeedsBuilding:boolean

+Country():void
+Country(states: ArrayList<State>):void
+findStateOrAdd(name: String): State
+addState(state: State): void
+getPersonList(): PersonList
+getTeamLinkedHashMap(): TeamLinkedHashMap
+getStates(): ArrayList<State>
+sort(): void
+addTeam(Team): void
+addTeams(ArrayList<Team>): void

---

**State**

-cities: ArrayList<City>
+isSorted: boolean
-stateName: String

+State(): void
+State(cities: ArrayList<City>):void
+findCityOrAdd(cityName: String):City
+addCity(city: City): void
+compareTo(state: State):int
+sort(): void
+getName():String
+getCities(): ArrayList<City>

---

**City**

-personList: PersonList
-cityName: String
-teams: teamLinkedHashMap
-latitude
-longitude

+City():void
+City(personList: PersonList):void
+getPersonList(): PersonList
+compareTo(city: City): int
+getName(): String
+getCoordinates(latitude: int, longitude:int)
+setLatitude(coordinate: int): void
+setLongitude(coordinate: int): void
+addTeam(Team: team): void

---

**PersonList**

-people: ArrayList<Person>
+firstNameComparator: CompareFirstName
+middleNamesComparator: CompareMiddleNames
+lastNameComparator: CompareLastName
+birthDateComparator: CompareBirthDate
+currentlySortedBy: String

+PersonList(): void
+getPerson(i:int): Person
+sortUsing(comparator: Comparator<Person>): void
+exactSearchFor(key: String, comparator: Comparator<Person>): PersonList
+partialSearchNamesFor(name: String): PersonList
+combineWith(personList: PersonList): PersonList
+addPerson(person: Person): void
+getPersonList(): ArrayList<Person>
+makeIntoChart(graphType: int): void

---

**Person**

-firstName: String
-lastName: String
-middleNames: String
-birthDate: Date
-cityName: String
-stateName: String

+Person(firstName:String, lastName:String, birthDate: Date)
+Person(firstName:String, lastName:String, middleNames: String, birthDate: Date)
+Person(fullName: String, birthDate: Date)
+getFirstName(): String
+getLastName(): String
+getMiddleNames(): String
+getBirthDate(): Date
+compareTo(otherPerson: Person): int
+getFullName(): String
+getCityName(): String
+getStateName(): String
+setCityName(): String
+setStateName(): String

---

**Pie extends JComponent**

-ageArray: int[]
-weightSum: int
-teamName: String
-sectors: ArrayList<Sector>

+Pie(personList: PersonList, teamName: String> map): void
+Pie(): void
+Pie(Team): void

---

**Sector extends Arc2D.Double**

-weight: int
-String: name
-Color: color

+Sector(name: String, weight: int)
+getWeight(): int
+getName(): String
+getColor(): Color
+setColor(color: Color): void

---

**TeamLinkedHashMap extends LinkedHashMap<String, Team>**

+addTeam(Team)
+sort()

---

**CustomBufferedReader extends BufferedReader**

CustomBufferedReader(): void
CustomBufferedReader(arg0: reader): void

---

**Team**

-stateName: String
-cityName: String
-teamMembers: PersonList

+getCityName(): String
+getStateName(): String
+getPersonList(): PersonList
+setCityName(): void
+setStateName(): void
+setPersonList():void

---

**<<Interface>>**
**Comparator<T>**

+compare(object1: t, object2: t): int

---

**CompareLastName**

+CompareLastName(): void
+compare(person1: Person, person2: Person): int

---

**CompareBirthDate**

+CompareBirthDate(): void
+compare(person1: Person, person2: Person): int

---

**CompareMiddleNames**

+CompareMiddleNames(): void
+compare(person1: Person, person2: Person): int

---

**CompareFirstName**

+CompareFirstName(): void
+compare(person1: Person, person2: Person): int

Text