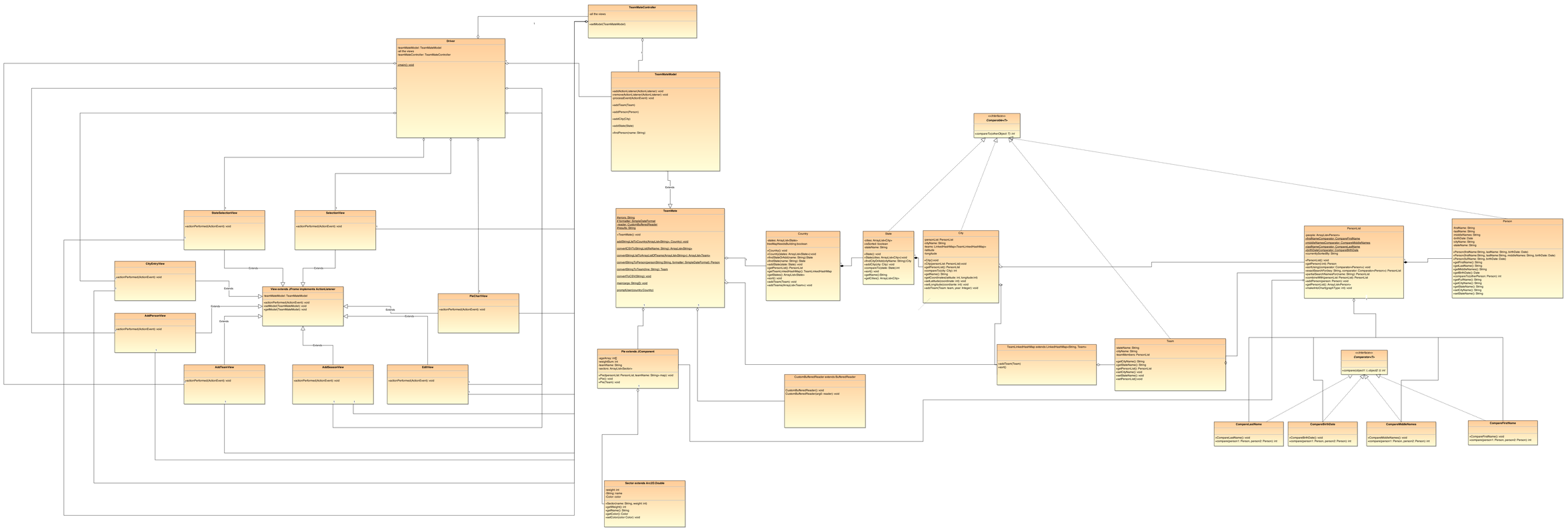
TeamMateController
all the views
+setModel(TeamMateModel)
Driver
teamMateModel: TeamMateModel
all the views
teamMateController: TeamMateController
+main() :void
TeamMateModel
+addActionListener(ActionListener): void
+removeActionListener(ActionListener): void
processEvent(ActionEvent): void
+addTeam(Team)
+addPerson(Person)
+addCity(City)
+addState(State)
+findPerson(name: String)
Extends
StateSelectionView
+actionPerformed(ActionEvent): void
SelectionView
+actionPerformed(ActionEvent): void
CityEntryView
+actionPerformed(ActionEvent): void
View extends JFrame implements ActionListener
teamMateModel: TeamMateModel
+actionPerformed(ActionEvent): void
+setModel(TeamMateModel): void
+getModel(TeamMateModel): void
PieChartView
+actionPerformed(ActionEvent): void
AddPersonView
+actionPerformed(ActionEvent): void
AddTeamView
+actionPerformed(ActionEvent): void
AddSeasonView
+actionPerformed(ActionEvent): void
EditView
+actionPerformed(ActionEvent): void
TeamMate
Person: String
formatter: SimpleDateFormat
reader: CustomBufferedReader
Results: String
+TeamMate() :void
addStringLeftToCountry(ArrayList<String>, Country): void
convertCSVToStringLeftSeName: String): ArrayList<String>
convertStringLeftToArrayLeftOfTeams(ArrayList<String>): ArrayList<Team>
convertStringToPerson(person:String, formatter: SimpleDateFormat): Person
convertStringToTeam(time: String): Team
convertToCSV(String): void
main(args: String[]): void
printsOutCountry(Country)
Country
states: ArrayList<State>
teaMayNeedsBuilding:boolean
+Country() :void
+Country(states: ArrayList<State>):void
+findStateOrAdd(name: String):State
+getStateIndex(state: State): int
+addStateIndex(state: State): void
+getPersonList(): PersonList
+getTeamLinkedHashMap(): TeamLinkedHashMap
+getStates(): ArrayList<State>
+setTeam(): void
+addTeamTeam(): void
+addTeamsArray(ArrayList<Team>): void
State
+cities: ArrayList<City>
+isSorted: boolean
-stateName: String
+State() :void
+State(cities: ArrayList<City>):void
+findCityOrAdd(name: String):City
+addCity(city: City): void
+compareToState: State):int
+sort() :void
+getName():String
+getCities(): ArrayList<City>
City
-personList: PersonList
-cityName: String
-teams: LinkedHashMap<Team,LinkedHashMap>
-latitude
-longitude
+City() :void
+City(personList: PersonList):void
+getPersonList(): PersonList
+compareTo(cities: City): int
+getName():String
+getCoordinates(latitude: int, longitude:int)
+setLatitude(coordinate: int): void
+setLongitude(coordinate: int): void
+addTeamTeam team, year: Integer): void
<<Interface>>
Comparable<T>
+compareTo(anotherObject: T):-int
PersonList
people: ArrayList<Person>
#firstNameComparator: CompareFirstName
#middleNameComparator: CompareMiddleNames
#lastNameComparator: CompareLastName
#birthDateComparator: CompareBirthDate
+currentlyAcceptBy: String
+PersonList() :void
+getPerson(): int): Person
+sortUsing(comparator: Comparator<Person>): void
+sortUsingCurrentComparator: Comparator<Person>): PersonList
+searchSearchNameEqualsName: String): PersonList
+searchWithPersonsCityEqualsCity: PersonList
+addPerson(person: Person): void
+getPersonList(): ArrayList<Person>
+massInsertOfkindgivenType: 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(firstName: String, birthDate: Date)
+getFirstName(): String
+getLastName(): String
+getMiddleNames(): String
+getBirthDate(): Date
+compareTo(otherPerson: Person): int
+setFirstName(): String
+setCityName(): String
+setStateName(): String
+setStateName(): String
TeamLinkedHashMap extends LinkedHashMap<String, Team>
+addTeam(Team)
+sort()
Team
-stateName: String
-cityName: String
-teamMembers: PersonList
+getCityName(): String
+getStateName(): String
+getPersonList(): PersonList
+setCityName(): void
+setStateName(): void
+setPersonList():void
<<InterfaceInv>>
Comparator<T>
+compare(object1: (object2: ()): 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
Pie extends JComponent
-angleArray: int[]
-weightSum: int
-teamName: String
-sectors: ArrayList<Sector>
+Pie(personList: PersonList, teamName: String=map): void
+Pie(): void
+Pie(Team): void
CustomBufferedReader extends BufferedReader
CustomBufferedReader() :void
CustomBufferedReader(arg0: reader): void
Sector extends Arc2D.Double
-weight: int
-Color color
-Color color
+ (Sector(name: String, weight: int)
+getWeight(): int
+getName(): String
+getColor(): Color
+setColor(color Color): void