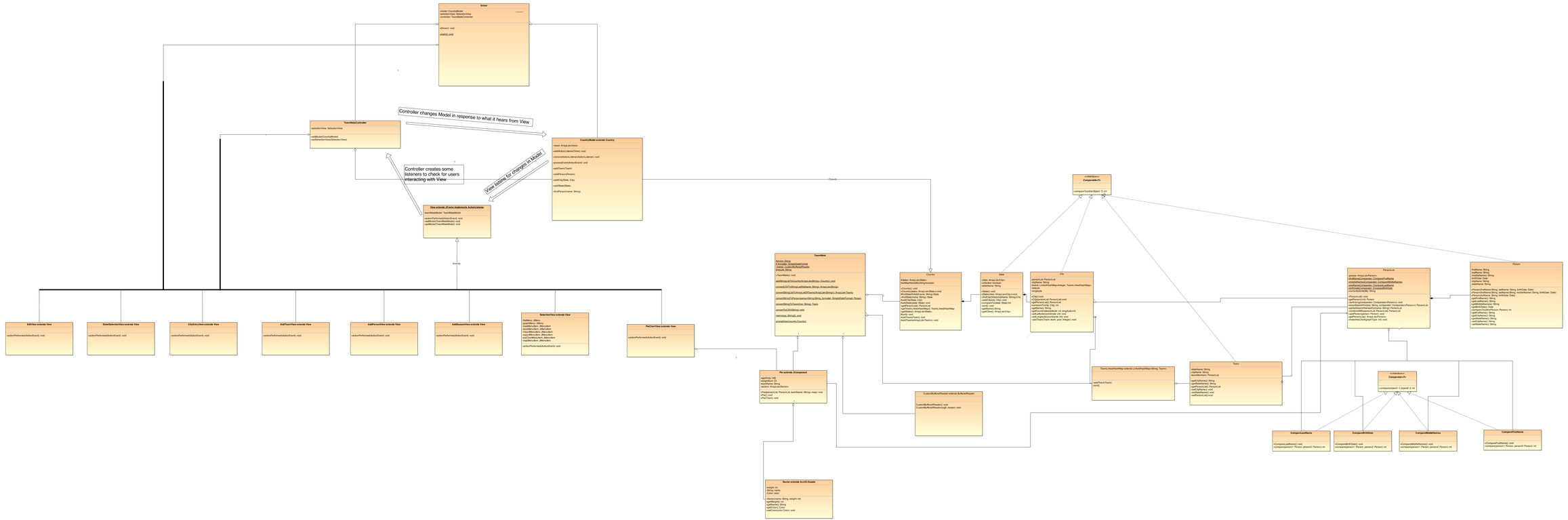
Driver
model: CountryModel
selectorView: SelectorView
controller: TeamMakeController
+Driver() :void
+main() :void
TeamMakeController
-selectorView: SelectorView
+setModel(CountryModel)
+setSelectorView(SelectorView)
Controller changes Model in response to what it hears from View
CountryModel extends Country
-views: ArrayList<View>
+addActionListener(View) :void
+removeActionListener(ActionListener) :void
+processEvent(ActionEvent) :void
+addTeam(Team)
+addPerson(Person)
+addCity(State, City)
+setState(State)
+delPerson(name: String)
Controller creates some listeners to check for users interacting with View
View listens for changes in Model
View extends JFrame implements ActionListener
#controlMenuItem: TeamMenuItem
+actionPerformed(ActionEvent) :void
+addMenuTeamMenuItem() :void
+getMenuTeamMenuItem() :void
extends
EditView extends View
+actionPerformed(ActionEvent) :void
StateSelectorView extends View
+actionPerformed(ActionEvent) :void
CityEntryView extends View
+actionPerformed(ActionEvent) :void
AddTeamView extends View
+actionPerformed(ActionEvent) :void
AddPersonView extends View
+actionPerformed(ActionEvent) :void
AddSeasonView extends View
+actionPerformed(ActionEvent) :void
SelectorView extends View
-frameLeft: JMenu
-frameRight: JMenu
-loadMenuItem: JMenuItem
-saveMenuItem: JMenuItem
-exportMenuItem: JMenuItem
-importMenuItem: JMenuItem
-addMenuItem: JMenuItem
-delMenuItem: JMenuItem
+actionPerformed(ActionEvent) :void
PieViewView extends View
+actionPerformed(ActionEvent) :void
Extends
Extends
TeamMate
#fanno: String
# formatter: SimpleDateFormat
# fanDate: SimpleDateFormat
#fanafs: String
+TeamMate() :void
addMinor(JsfCountry,teamname: String, Country) :void
convertCSVToMinor(setfileName: String, ArrayList<Minor>)
convertMinorLsfToArrayLsfAllTeamsArrayLsfs(Minor[] ,ArrayLsfs<Team>)
convertMinorLsfToarrayLsfs(Minor[]:String, formatter: SimpleDateFormat): Person
convertMinorLsfTeamline: String): Team
convertToCSV(Minor): void
+toArrayLsfs(Minor[]): void
+toArrayLsfs(countryLsf:Country)
Country
#states: ArrayList<State>
#isAllNeededAdding:boolean
-name: String
#Country() :void
#Country(pstate: ArrayList<State>) :void
#Country(pstate: ArrayList<City>) :void
#initDefaultsArray(fsfname: String,City
+getCities() :ArrayList
+setAllCities(ArrayLsfCities: ArrayLsfCities)
+getAllCities() :ArrayLsfCities
+getCities(State) :ArrayLsfCities
+addCities(City, State) :void
+addCities(City) :void
+addCities(ArrayLsfCities) :void
State
-cities: ArrayLsfCity>
-isSelected: boolean
-stateName: String
+State() :void
+State(pcities: ArrayLsfCity>) :void
+initCityArrayArrayfsfname: String,City
+getCities() :ArrayLsfCity>
+setAllCities(ArrayLsfCities: State) :void
+getCities(State) :ArrayLsfCities
+getName() :String
+getCities() :ArrayLsfCity>
City
-present:lsf: PersonLsf
-cityName: String
-teamsMap: TeamLinkedHashMap<Integer, TeamLinkedHashMap>
-stateLsfs
-cityLsfs
+City() :void
+City(pnamelsf: PersonLsf) :void
+getPersonLsf() :PersonLsf
+setPersonLsf(pnamelsf: PersonLsf) :void
+getName() :String
+getCoordinateMatrix(fsfx: int,fsfy:int) :int[][]
+setCityLsf(pcityLsf: CityLsf) :void
+setLongitude(pcoordinate: int) :void
+addTeam(fsfTeam: team, year: Integer) :void
<<interface>>
Comparator<T>
+compareTo(fsfarrObject: T) :int
Person
-firstName: String
-lastName: String
-middleNames: String
-birthDate: Date
-deathDate: Date
-stateName: String
+Person() :void
+Person(firstNameParm: String, lastNameParm: String, birthDate: Date)
+Person(firstNameParm: String, lastNameParm: String, middleNames: String, birthDate: Date)
+getFirstName() :String
+setFirstName(): String
+getLastName() :String
+setLastName(): String
+getBirthDate() :Date
+setBirthDate(pbirthDate: int): int
+getFullName() :String
+setFullName(): String
+getDeathDate() :Date
+setDeathDate(): String
+getStateName() :String
+setStateName(): String
PersonLsf
-people: ArrayLsfs<Person>
-firstNameComparator: ComparatorFirstName
-middleNameComparator: ComparatorMiddleNames
-lastNameComparator: ComparatorLastName
-birthDateComparator: ComparatorBirthDate
-deathDateComparator: String
+PersonLsf() :void
+getPersonLsf() :Person
+setAllPeopleLsf(people: Comparator<Person>) :void
+getPeopleLsf(pcomparator: Comparator<Person>): Person[]
+getPeopleLsf(pfirstNameparm: String): PersonLsf
+getPeopleLsf(pmiddleNameparm: String): PersonLsf
+getPeopleLsf(plastNameparm: String): PersonLsf
+getPeopleLsf(): ArrayLsfs<Person>
+makeAllNeededTeamsLsf(fsfpersonLsf): void
ComparatorLastName
+Compare(person1: Person, person2: Person): int
ComparatorBirthDate
+Compare(Date[]): void
+compare(person1: Person, person2: Person): int
ComparatorMiddleNames
+Compare(Middle[]): void
+compare(person1: Person, person2: Person): int
ComparatorFirstName
+Compare(Fnlist[]): void
+compare(person1: Person, person2: Person): int
Pie extends JComponent
-appArray: int[]
-appLabel: int[]
-teamName: String
-sectors: Array<Sector>
+Pie(Persons:Lsf, Persons:Lsf, teamName: String<map>) :void
+Pie() :void
+Pie(Team): void
CustomBufferedReader extends BufferedReader
CustomBufferedReader() :void
CustomBufferedReader(arg0: reader): void
TeamLinkedHashMap extends LinkedHashMap<String, Team>
+addTeamsTeam()
+setlf()
Team
-teamName: String
-teamLsf: PersonLsf
-teamMembers: PersonLsf
+getFullName() :String
+getStateName() :String
+getPersonLsf() :PersonLsf
+setPersonLsf(): void
+setStateName(): void
+setPersonLsf(): void
Sector extends Arc2D.Double
-weight: int
-String name
-int[] color
+Sector(name: String, weight: int)
+getWeight() :int
+setWeight(): void
+getColor() :Color
+setColor(color Color): void