Controller creates some listeners to check for users interacting with View +setSportsStuff(SportsStuff):void +removeActionListener(ActionListener): void <<Interface>>
Comparable<T> -processEvent(ActionEvent): void +prepareStatesFromTextFile(String):void package otherClasses +FileChooserView(actionListener: ActionListener) PersonList implements Serializable

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

+PersonList(): void
+getPerson(i:nt): Person
+getPeople:ArrayList<Person>
+getPersonList(): ArrayList<Person>
+getPersonList(): ArrayList<Person>): void
+exactSearchFor(key: String, comparator<Person>): void
+exactSearchFor(key: String, comparator<PersonList
+partialSearchNamesFor(name: String): PersonList
~combineWith(personList: PersonList): PersonList
~addPerson(person: Person): void
+makeIntoChart(graphType: int): void
+returnAfterSortingUsing(Comparator<Person>): PersonList
+toString(): String +DegreesOfSeparationView(): void +actionPerformed(ActionEvent): void -lastName: String
-middleNames: String
-birthDate: Date
-cityName: String
-stateName: String
-stateName: String
-age: int
-HashMap<TeamSeason> seasons;

+Person(firstName:String, lastName: String, birthDate: Date):void
+Person(firstName: String, lastName: String, birthdate: Date, deathDate: Date):void
+Person(firstName:String, lastName:String, middleNames: String, birthDate: Date):void
+Person(fullName:String, birthDate: Date):void
+Person(fullName:String):void SelectionView extends View -states: ArrayList<State>
-allTheCities: ArrayList<City>
-allThePeople: ArrayList<City>
-allThePeople: ArrayList<City>
-allThePeople: ArrayList<City>
-allThePeople: ArrayList<City>
-allThePeople: ArrayList<City>
-allThePeople: ArrayList<State>
-yestOuntry(): void
+Country(): void
+Country(): void
+Country(): void
+Country(): Void
+Country(): Void
+Country(): Void
+GetPersonList(): PersonList
+getTeamLinkedHashMap(): TeamLinkedHashMap
+getStates(): ArrayList<State>
+getCities(): ArrayList<Cities>
+getPeople(): ArrayList<Person>
+getSportStuff(): SportsStuff

+findCity(State, Sting): City
+findState(String): State
+findPerson(String): Person

#findStateOrAdd(name: String):State

#addCity(State, City)
#addState(state: State): void
#addTeam(Team): void
#addTeam(ArrayList<Team>): void
#addPerson(Person): void
#addPerson(City, Person): void
#removePerson(Persons person): void
#removePerson(Persons person): void
#removeCity(City city): void
#sort(): void
+toString(): String -cities: ArrayList<City>
-allThePeopleInTheState: ArrayList<Person>
+isSorted: boolean
-stateName: String

+State(): void
+State(cities: ArrayList<City>):void

+getName():String
+getCities(): ArrayList<City>

~findCityOrAdd(cityName: String):City
~addCity(city: City): void

+compareTo(state: State):int
~sort(): void PersonEntryView extends View -enterButton: JButton
-city: City
personName: JTextField
birthDate: JTextField
deathDate: JTextField

+PersonEntryView(aCity: City):void
+getEnterButton(): JButton
+getFullPerson(): Person
+getCity(): City
+actionPerformed(ActionEvent): void -state: State
cityEntryJTextField: JTextField
latitudeEntryJTextField: JTextField
longitudeEntryJTextField: JTextField
enterButton: JButton
+CityEntryView(state: State): void
+getEnterButton(): JButton
+getFullCity(): City
+actionPerformed(ActionEvent): void
+closeWindow(): void -enterButton:JButton
+TeamEntryView(): void
+actionPerformed(ActionEvent): void SeasonEntryView extends View +City(cityName:String, state: State): void +getPersonList(): PersonLlst +getName(): String +getCoordinate(): Point2d +getState(): State +getFirstName(): String
+getLastName(): String
+getMiddleNames(): String
+getBirthDate(): String
+getBirthDate(): Date
+getCityName(): String
+getStateName(): String
+getAge(): int
+getAge() int
+getAge(Date, Date): int

~setDeathDate(Date):void
~setCityName(String): void
~setStateName(String): void
~addSeason(TeamSeason season)

+compareTo(otherPerson: Person): int
+compareTo(Person):int
+convertStringToPerson(String, SimpleDateFormat): Person +SelectionView():void
+getLoadMenu(): JMenuItem
+getSaveMenu(): JMenuItem
+getImportMenuItem(): JMenuItem
+getExportMenuItem(): JMenuItem
+getPieChartMenuItem(): JMenuItem
+getMapMenuItem(): JMenuItem
+getAddPersonButton(): JButton
+getAddPlaceButton(): JButton
+getAddTeamButton(): JButton
+getJList(): JList<City>
+setModel(CountryModel): void
+actionPerformed(ActionEvent): void
-updatePlaceList(): void
-updatePersonList(): void
-updateSeasonList(): void +PieChartView(pie: Pie):void +actionPerformed(ActionEvent): void ~addTeam(Team: team, year: Integer): void
~addPerson(Person):void
~addCity(City):void
+compareTo(city: City): int Pie extends JComponent -ageArray: int[]
-weightSum: int
-teamName: String
-sectors: ArrayList<Sector>

+Pie(tpersonLlst: PersonList, teamName: String): void
+Pie(): void
+Paint(g: Graphics): void Sector extends Arc2D.Double +compare(object1: t, object2: t): int +Sector(name: String, weight: int)
+getWeight(): int
+getName(): String
+getColor(): Color
+setColor(color Color): void +CompareFirstName(): void +compare(person1: Person, person2: Person): int +CompareStateThenCity(): void +compare(state1: State, state2: State): int +CompareLastName(): void +compare(person1: Person, person2: Person): int +CompareBirthDate(): void +compare(person1: Person, person2: Person): int +CompareMiddleNames(): void +compare(person1: Person, person2: Person): int +CompareCityThenState(): void +compare(city1: City, city2: Person): int package sports SportsStuff implements Serializable

-teams: TreeMap<String, Team>
-teamSeasons: ArrayLlst<TeamSeason>

+SportsStuff()
-naddTeam(String, Team):void
+addSeason(String, TeamSeason, int):void
+prepareFromUsingCountryModel(String, CounttyModel):void
+findOrAddSportsYearsFromInt(int): SportsYear

package MVCStuff

package countryComponents