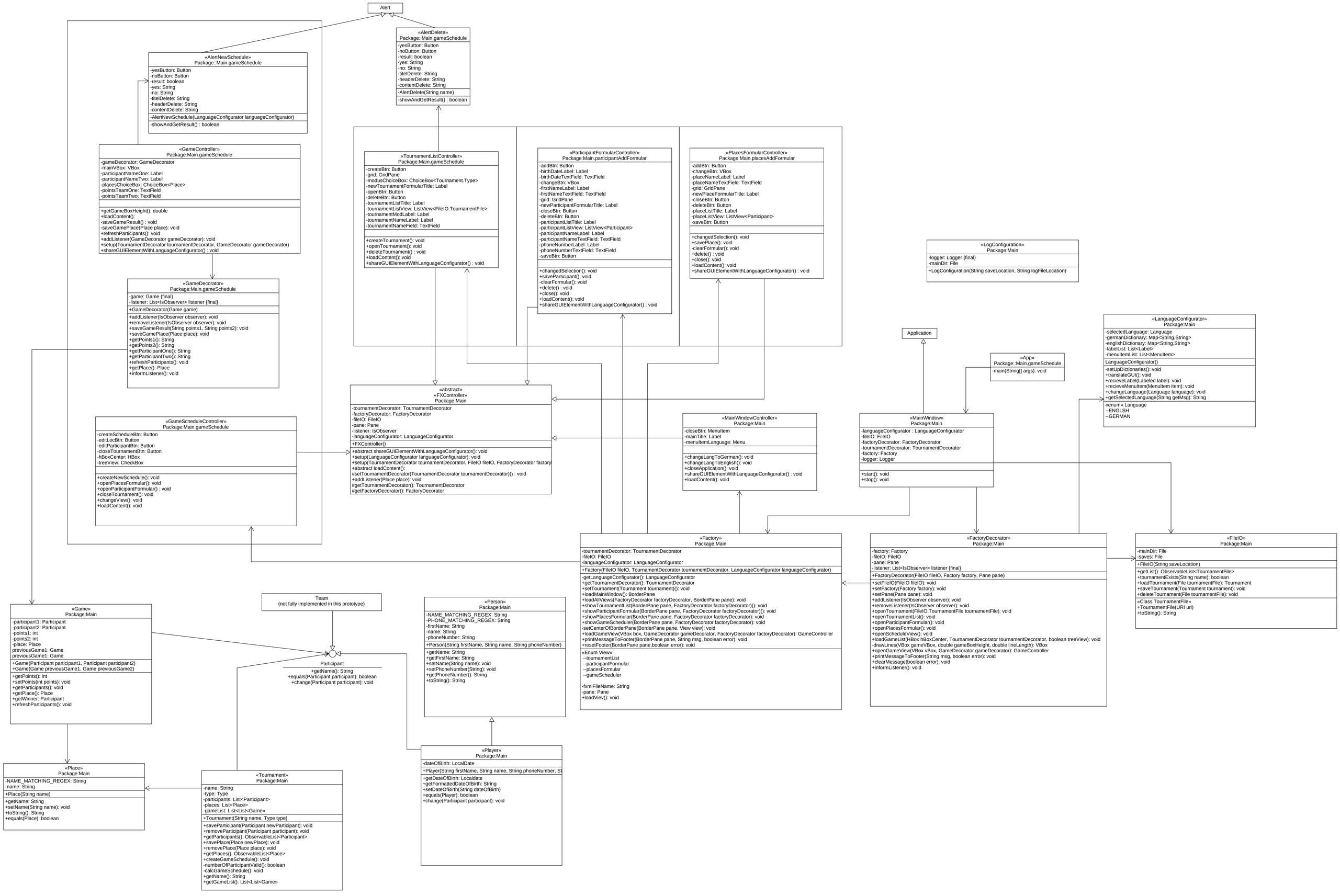
Alert
«AlertNewSchedule»
Package::Main.gameSchedule
-yesButton: Button
-noButton: Button
-result: boolean
-yes: String
-no: String
-titelDelete: String
-headerDelete: String
-contentDelete: String
-AlertNewSchedule(LanguageConfigurator languageConfigurator)
-showAndGetResult() : boolean
«AlertDelete»
Package::Main.gameSchedule
-yesButton: Button
-noButton: Button
-result: boolean
-yes: String
-no: String
-titelDelete: String
-headerDelete: String
-contentDelete: String
-AlertDelete(String name)
-showAndGetResult() : boolean
«GameController»
Package:Main.gameSchedule
-gameDecorator: GameDecorator
-mainVBox: VBox
-participantNameOne: Label
-participantNameTwo: Label
-placesChoiceBox: ChoiceBox<Place>
-pointsTeamOne: TextField
-pointsTeamTwo: TextField
+getGameBoxHeight(): double
+loadContent():
-saveGameResult() : void
-saveGamePlace(Place place): void
+refreshParticipants(): void
+addListener(GameDecorator gameDecorator): void
+setup(TournamentDecorator tournamentDecorator, GameDecorator gameDecorator)
+shareGUIElementWithLanguageConfigurator() : void
«TournamentListController»
Package:Main.gameSchedule
-createBtn: Button
-grid: GridPane
-modusChoiceBox: ChoiceBox<Tournament.Type>
-newTournamentFormularTitle: Label
-openBtn: Button
-deleteBtn: Button
-tournamentListTitle: Label
-tournamentListView: ListView<FileIO.TournamentFile>
-tournamentModLabel: Label
-tournamentNameLabel: Label
-tournamentNameField: TextField
+createTournament(): void
+openTournament(): void
+deleteTournament() : void
+loadContent(): void
+shareGUIElementWithLanguageConfigurator() : void
«ParticipantFormularController»
Package:Main.participantAddFormular
-addBtn: Button
-birthDateLabel: Label
-birthDateTextField: TextField
-changeBtn: VBox
-firstNameLabel: Label
-firstNameTextField: TextField
-grid: GridPane
-newParticipantFormularTitle: Label
-closeBtn: Button
-deleteBtn: Button
-participantListTitle: Label
-participantListView: ListView<Participant>
-participantNameLabel: Label
-participantNameTextField: TextField
-phoneNumberLabel: Label
-phoneNumberTextField: TextField
-saveBtn: Button
+changedSelection(): void
+savePartcipant(): void
-clearFormular(): void
+delete() : void
+close(): void
+loadContent(): void
+shareGUIElementWithLanguageConfigurator() : void
«PlacesFormularController»
Package:Main.placesAddFormular
-addBtn: Button
-changeBtn: VBox
-placeNameLabel: Label
-placeNameTextField: TextField
-grid: GridPane
-newPlaceFormularTitle: Label
-closeBtn: Button
-deleteBtn: Button
-placeListTitle: Label
-placeListView: ListView<Participant>
-saveBtn: Button
+changedSelection(): void
+savePlace(): void
-clearFormular(): void
+delete() : void
+close(): void
+loadContent(): void
+shareGUIElementWithLanguageConfigurator() : void
«LogConfiguration»
Package:Main
-logger: Logger {final}
-mainDir: File
+LogConfiguration(String saveLocation, String logFileLocation)
«GameDecorator»
Package:Main.gameSchedule
-game: Game {final}
-listener: List<IsObserver> listener {final}
+GameDecorator(Game game)
+addListener(IsObserver observer): void
+removeListener(IsObserver observer): void
+saveGameResult(String points1, String points2): void
+saveGamePlace(Place place): void
+getPoints1(): String
+getPoints2(): String
+getParticipantOne(): String
+getParticipantTwo(): String
+refreshParticipants(): void
+getPlace(): Place
+informListener(): void
«abstract»
«FXController»
Package:Main
-tournamentDecorator: TournamentDecorator
-factoryDecorator: FactoryDecorator
-fileIO: FileIO
-pane: Pane
-listener: IsObserver
-languageConfigurator: LanguageConfigurator
+FXController()
+abstract shareGUIElementWithLanguageConfigurator(): void
+setup(LanguageConfigurator languageConfigurator): void
+setup(TournamentDecorator tournamentDecorator, FileIO fileIO, FactoryDecorator factory
+abstract loadContent():
#setTournamentDecorator(TournamentDecorator tournamentDecorator)() : void
+addListener(Place place): void
#getTournamentDecorator(): TournamentDecorator
#getFactoryDecorator(): FactoryDecorator
«MainWindowController»
Package:Main
-closeBtn: MenuItem
-mainTitle: Label
-menuItemLanguage: Menu
+changeLangToGerman(): void
+changeLangToEnglish(): void
+closeApplication(): void
+shareGUIElementWithLanguageConfigurator() : void
+loadContent(): void
Application
«App»
Package::Main.gameSchedule
-main(String[] args): void
«LanguageConfigurator»
Package:Main
-selectedLanguage: Language
-germanDictionary: Map<String,String>
-englishDictionary: Map<String,String>
-labelList: List<Label>
-menuItemList: List<MenuItem>
LanguageConfigurator()
-setUpDictionaries(): void
+translateGUI(): void
+recieveLabel(Labeled label): void
+recieveMenuItem(MenuItem item): void
+changeLanguage(Language language): void
+getSelectedLanguage(String getMsg): String
«enum» Language
--ENGLSH
--GERMAN
«MainWindow»
Package:Main
-languageConfigurator : LanguageConfigurator
-fileIO: FileIO
-factoryDecorator: FactoryDecorator
-tournamentDecorator: TournamentDecorator
-factory: Factory
-logger: Logger
+start(): void
+stop(): void
«GameScheduleController»
Package:Main.gameSchedule
-createScheduleBtn: Button
-editLocBtn: Button
-editParticipantBtn: Button
-closeTournamentBtn: Button
-hBoxCenter: HBox
-treeView: CheckBox
+createNewSchedule(): void
+openPlacesFormular(): void
+openParticipantFormular() : void
+closeTournament(): void
+changeView(): void
+loadContent(): void
«Factory»
Package:Main
-tournamentDecorator: TournamentDecorator
-fileIO: FileIO
-languageConfigurator: LanguageConfigurator
+Factory(FileIO fileIO, TournamentDecorator tournamentDecorator, LanguageConfigurator languageConfigurator)
-getLanguageConfigurator(): LanguageConfigurator
+getTournamentDecorator(): TournamentDecorator
+setTournament(Tournament tournament)(): void
+loadMainWindow(): BorderPane
+loadAllViews(FactoryDecorator factoryDecorator, BorderPane pane): void
+showTournamentList(BorderPane pane, FactoryDecorator factoryDecorator)(): void
+showParticipantFormular(BorderPane pane, FactoryDecorator factoryDecorator)(): void
+showPlacesFormular(BorderPane pane, FactoryDecorator factoryDecorator): void
+showGameScheduler(BorderPane pane, FactoryDecorator factoryDecorator): void
-setCenterOfBorderPane(BorderPane pane, View view): void
+loadGameView(VBox box, GameDecorator gameDecorator, FactoryDecorator factoryDecorator): GameController
+printMessageToFooter(BorderPane pane, String msg, boolean error): void
+resetFooter(BorderPane pane,boolean error): void
«Enum View»
--tournamentList
--participantFormular
--placesFormular
--gameScheduler
-fxmlFileName: String
-pane: Pane
+loadViev(): void
«FactoryDecorator»
Package:Main
-factory: Factory
-fileIO: FileIO
-pane: Pane
-listener: List<IsObserver> listener {final}
+FactoryDecorator(FileIO fileIO, Factory factory, Pane pane)
+setFileIO(FileIO fileIO): void
+setFactory(Factory factory): void
+setPane(Pane pane): void
+addListener(IsObserver observer): void
+removeListener(IsObserver observer): void
+openTournament(FileIO.TournamentFile tournamentFile): void
+openTournamentList(): void
+openParticipantFormular(): void
+openPlacesFormular(): void
+openScheduleView(): void
+loadGameList(HBox hBoxCenter, TournamentDecorator tournamentDecorator, boolean treeView): void
-drawLines(VBox gameVBox, double gameBoxHeight, double lineLength): VBox
+openGameView(VBox vBox, GameDecorator gameDecorator): GameController
+printMessageToFooter(String msg, boolean error): void
+clearMessage(boolean error): void
+informListener(): void
«FileIO»
Package:Main
-mainDir: File
-saves: File
+FileIO(String saveLocation)
+getList(): ObservableList<TournamentFile>
+tournamentExists(String name): boolean
+loadTournament(File tournamentFile): Tournament
+saveTournament(Tournament tournament): void
+deleteTournament(File tournamentFile): void
«Class TournamentFile»
+TournamentFile(URI uri)
+toString(): String
«Game»
Package:Main
-participant1: Participant
-participant2: Participant
-points1: int
-points2: int
-place: Place
previousGame1: Game
previousGame1: Game
+Game(Participant participant1, Participant participant2)
+Game(Game previousGame1, Game previousGame2)
+getPoints(): int
+setPoints(int points): void
+getParticipants(): void
+getPlace(): Place
+getWinner: Participant
+refreshParticipants(): void
Team
(not fully implemented in this prototype)
Participant
+getName(): String
+equals(Participant participant): boolean
+change(Participant participant): void
«Person»
Package:Main
-NAME_MATCHING_REGEX: String
-PHONE_MATCHING_REGEX: String
-firstName: String
-name: String
-phoneNumber: String
+Person(String firstName, String name, String phoneNumber)
+getName: String
+getFirstName: String
+setName(String name): void
+setPhoneNumber(String): void
+getPhoneNumber(): String
+toString(): String
«Place»
Package:Main
-NAME_MATCHING_REGEX: String
-name: String
+Place(String name)
+getName: String
+setName(String name): void
+toString(): String
+equals(Place): boolean
«Tournament»
Package:Main
-name: String
-type: Type
-participants: List<Participant>
-places: List<Place>
-gameList: List<List<Game»
+Tournament(String name, Type type)
+saveParticipant(Participant newParticipant): void
+removeParticipant(Participant participant): void
+getParticipants(): ObservableList<Participant>
+savePlace(Place newPlace): void
+removePlace(Place place): void
+getPlaces(): ObservableList<Place>
+createGameSchedule(): void
-numberOfParticipantValid(): boolean
-calcGameSchedule(): void
+getName(): String
+getGameList(): List<List<Game»
«Player»
Package:Main
-dateOfBirth: LocalDate
+Player(String firstName, String name, String phoneNumber, St
+getDateOfBirth: Localdate
+getFormattedDateOfBirth: String
+setDateOfBirth(String dateOfBirth)
+equals(Player): boolean
+change(Participant participant): void