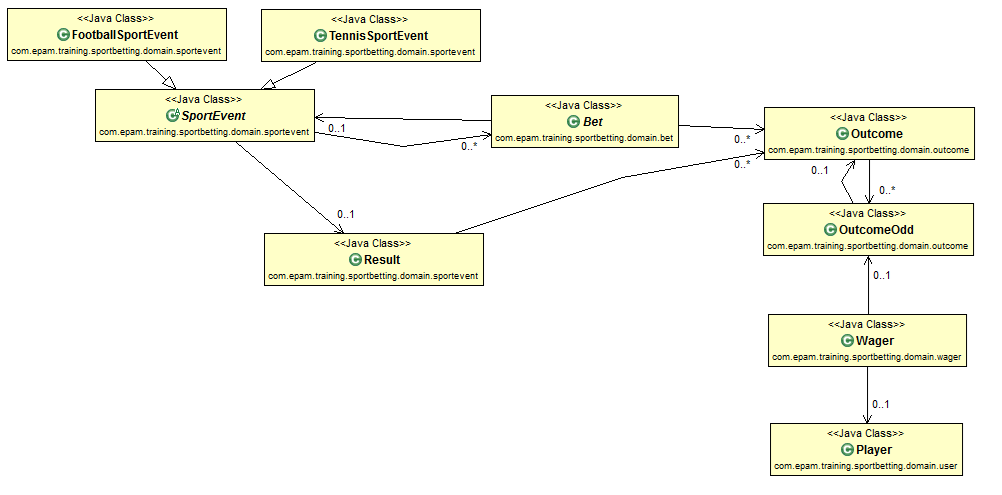
## OO-1 - Domain Model for the Sports Betting Application

**Conversation**

Sports betting is the activity of predicting sports results and placing a wager on the outcome.

Implement the domain model for the Sports Betting application.

The betting domain:



**Sport event**

Title

Start date

End date

Different types (football, tennis), use inheritance

**Bet**: a bet has possible outcomes, outcomes can be predicted with wagers by the players

**Sport event**

Description

Different **outcomes** of the bet (list)

Different types of the bet (betting for goals, winner, player’s score), use an enum property

**Outcome**: a possible outcome of a bet; for example on a winner bet it is the winner teams name in the value, on a goals bet it is the number of the goals in the value

Value (outcome)

**Outcome odds**: Outcome odd can be changed in time, but if the odd changed, a new outcome odd record must be created and the same time we can have only one valid odd (no overlapping of odds in time)

**Outcome odd**: the odd of an outcome; stands for how many times multiplied your money that you bet on a match result if you win

**Outcome**

Odd value

Valid from

Valid to

**Wager**: the wager placed by a Player on an outcome; It stores the odd, the amount and the currency of the bet, the date and time when the bet is created and the state of having been processed or not.

**Player**

**Outcome odd**

Amount

Currency

Timestamp

un-processed / processed

Win/Lose

**Player**: the player who places odds

Name

Account number

Balance

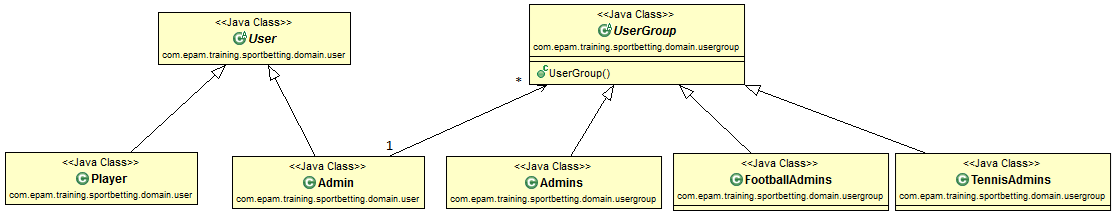
Currency (EUR, USD, HUF)

Date of birth

**Result**: multiple outcomes of a match, e.g.: 5 goals shot, the arsenal won, Oliver Giroud scored 2 goals

**Some examples for Events, Bets, Outcomes and Outcome odds can be found in the Appendix #1: Betting data**example**.**

The user domain:



**User**: the user of the application

Email (username)

Password

Enabled (Y/N)

**Player**: the player, who places odds

**Admin**: the administrators of the application

**User** **group**: groups of the admins

**Admin**: admin to modify all outcome odds

**Football admin**: admin to modify football odds

**Tennis admin**: admin to modify tennis odds

**Note:**

In this task we will only use the Player from the user domain.

**Task:**

Create a command-line based application with the following functions:

* Create the Java classes of the domain model
* When the program starts, it creates the following domain data with test (example) values:
  + Sport events
  + Bets
  + Outcomes
  + Outcome odds
  + Player data
* Functions to implement:
  + List all the bets with the possible outcomes
  + The user (player) can bet on games by choosing one possible outcome of a bet and specifying the amount they wish to wager
* When the user chooses to exit, random results are generated for all events, and the system calculates the user’s wagers result and prints it to the output.

The project name has to be “**sports-betting-application**”, and it has to contain at least the following classes:

|  |  |
| --- | --- |
| **Class** | **Description** |
| **com.epam.training.sportsbetting.App** | Entry point for the application, controls the application flow, uses services and domain classes. |
| **com.epam.training.sportsbetting.domain.\*** | Domain objects (see above) |
| **com.epam.training.sportsbetting.service.\*** | Classes containing logic of the application (getting the list of bets, checking user balance, etc.) |
| **com.epam.training.sportsbetting.ui.\*** | Classes related to communication with the user (read and write console) |

You can create any number of additional classes if you wish to.

Hints

Apply REPL (<http://en.wikipedia.org/wiki/REPL>)

**Confirmation**

* At least 15 classes created
* All the possible bets listed
* Player can choose multiple bets
* Player cannot bet more money that their balance
* Randomize outcomes to result
* Update everything that depends on the result (calculate new balance, etc.)
* Print the new balance

**Optional**

* You can provide the Player their data on the console after it was entered

**Example output**

|  |
| --- |
| > Hi, what is your name?  John Doe  > What is your account number?  5464545-5165165  > How much money do you have (more than 0)?  250000  > What is your currency? (UAH, EUR or USD)  UAH  > When were you born? eg.:1990-02-03  1979-04-25  Welcome John Doe !  Your balance is 250.000 UAH  > Please choose an outcome to bet on! (choose a number or press q for quit)  > 1: Bet on the Arsenal vs Chelsea sport event, the player Oliver Giroud will score 1. The odd on this is 10.0, valid from 2016-02-03 to 2016-02-05  > 2: Bet on the Arsenal vs Chelsea sport event, the number of scored goals will be 3. The odd on this is 1.3, valid from 2016-02-03 to 2016-02-05  > 3: Bet on the Arsenal vs Chelsea sport event, the winner will be Arsenal. The odd on this is 4.0, valid from 2016-02-03 to 2016-02-05  1  > How much do you want to bet on it? (q for quit)  50000  > Your new balance is 200.000 UAH  > What are you want to bet on? (choose a number or press q for quit)  > 1: Bet on the Arsenal vs Chelsea sport event, the player Oliver Giroud will score 1. The odd on this is 10.0, valid from 2016-02-03 to 2016-02-05  > 2: Bet on the Arsenal vs Chelsea sport event, the number of scored goals will be 3. The odd on this is 1.3, valid from 2016-02-03 to 2016-02-05  > 3: Bet on the Arsenal vs Chelsea sport event, the winner will be Arsenal. The odd on this 4.0, valid from 2016-02-03 to 2016-02-05  2  > What amount do you wish to bet on it? (press q to quit)  250000  > You don't have enough money, your balance is 200.000 UAH  > What amount do you wish to bet on it? (press q to quit)  200000  > What amount do you wish to bet on it? (press q to quit)  q  > Results:  > The winner is Outcome 2 [value=3, outcomeOdds=1.3 and valid from 2016-02-03 to 2016-02-05]  > You have won 260.000 UAH |