**PART 1 -- MY SOLUTION**

The application consists of a package containing all the Java Beans, imported as a Jar File into another project that uses them to test that they respond as they should.

I start by searching for a random player and a random tournament and linking them. The player constructor only takes the ID and the name because the rest of the parameters, at the moment of creating a player, have a clear default value. The next thing done from the main is to create objects of type Message, Game, and Deferral, and declare all of them as listeners of the player.

Once this is done, we can proceed to the lines that actually test the Java Beans package. When changing the date of the next game for a player (or setting it to null), an event is emitted to its listeners. MessageBean displays the corresponding message, and GameBean either creates a new game or updates it to give it a result. In my solution, it takes into account whether the player already has a game and whether the new date value is null or not. If the player already has a game date set, you cannot introduce another date that is not null. Additionally, to introduce a null date, the player must have a pending game. It's done analogously in GameBean.

An event is also emitted when the deferral date of a player is changed. In this case, it takes into account whether the date value is null or not and whether the game result is "PENDING". Only if the date is not null and the game result is "PENDING" a new Deferral is generated and a message regarding this is inserted.

**PART 2 – DIFFICULTIES**

The main difficulty I encountered when carrying out this assessable task was understanding the logic of what was being asked of me. It took me a lot of effort to truly understand the conditions that were required, such as the need to change the state of the match before entering a null date or that it was necessary to change the status of the deferral before entering a null deferral date.

**PART 3 – POSSIBLE EXTENSIONS**

One possible extension that I find very interesting is to use Hibernate to access the database and also access files to log information and then display it on a web page.

All of this would involve combining everything studied in the course into a single practical project, and it wouldn't necessarily entail a very large effort, since much of the already written code can be reused.