Refactoring Project : Iteration 1

In this iteration we will try to provide some observations on how to make the code better and more readable respecting good practices of java coding.

First of all there is some general observations that apply to all classes :

* Respect naming conventions.
* Make the attributes private and add getters and setters for better encapsulation.
* Edit the constructors conveniently.
* Add @Override tag to : toString() Method.
* Add class AbstractDAO with a connection object that guarantees single connexion using the singleton design pattern.
* Create an Interface for the database access objects that should be created to seperate business logic from databse access.
* Commented-out code sections should be removed to remove confusion.
* Add comments and documentation to the code for better readability.
* Add a .properties file

1. Class Equipe :

* Name of the team is stored in eq1, hence there is no need to use the attribute eq2 as a string.

1. Class MatchM :

* Attributes eq1 and eq2 should be of type Equipe instead of string.
* Edit the constructor conveniently.
* Make match status enum type to include a third state : ongoing

1. Class Tournoi :

* Database access should be done with the help DAOs
* Use prepared statements to protect against SQL injections.
* Use Exception Handling with good practices in the DAO classes that should be created to acces database.
* String comparaisons could be done with other better methods, in example : instead of comparing a string with an empty one using ==, we should use isEmpty().
* Separating UI with business logic.
* Make ajouterTour() method more readable and more compact by breaking it down to smaller helper functions .
* Deal with duplicated code by creating seperated helper methods.
* There some numbers that are used in the SQL queries, it is better to make them constantes

1. Class Belote:

* Some strings like : (jdbc:hsqldb:file) and paths  should be defined in a properties file in order to make them configurable and not hard coded.
* Exception handling should yield more meaningful messages.
* Code duplication.
* Use DAOs
* Avoid usign System.exit(0) for a more graceful exit.