Mcq - Unit tests

MCQ Game Java application with GUI, unit tests part.

In this part, the goal is to implement **unit tests** for the `Question` class **only**.

Warning:

- Content of `Question` class source file, written in *Model* part and located in `/part1-Model/src/...` , has to be **copied/pasted** in the same file in `/part2-Unit-Tests/src/` .
- Unit tests must be written in `/part2-Unit-Tests/test` folder.
- `junit` library jars are located in `/part2-Unit-Tests/lib` folder.

1. Project configuration

Doing it

1. **Configure project** so that:
   - `/part2-Unit-Tests/src` is considered as *source*,
   - `/part2-Unit-Tests/test` is considered as *test*,
   - `Junit` jars (either *Junit5* or *JUnit3-4*) are added to project libraries.
   - any other directory (especially `/part1-Model/src`) is considered as *excluded*,
2. **Complete** `QuestionTest` source code, writing **three tests**:
   - the **first** test creates an object and verifies that getters return expected values,
   - the **second** test creates an object and verifies that `toString` method returns expected value.
   - the **third** test creates an object and verifies that `getAnswer` method returns `null` when `answerId` is out of bounds.

Checking it

- **Run tests** and check that all tests pass.

Committing/Pushing it

- **Commit** changes with `2-Unit-Tests` as message brief (and details if checking failed).
- **Push** immediately.
- **Check** that remote repository has been updated.