

Programming Applications with Databases

Exercise Set 10

1. [2p] Create a repository based on NHibernate (or a similar ORM library). Make sure the following functionalities are introduced:

- [1p] Identifiers based on HiLow generation (in case it is not supported in your ORM, please find a corresponding solution),
- [1p] Components,
- [1p] Enumerations,
- [1p] Collections,
- [1p] Associations.

In this exercise it is not required to implement filtering, sorting and paging in the "retrieve" methods, however the following methods are expected:

- *Find(int id)*, can be based on the structure like *session.Get<User>(id)*, and
- *FindAll()*, can be based on the structure like *session.CreateQuery("from User").List<User>()*.

Finally, configure the solution to use the new repository and create (at least) a simple command line client to execute the use-cases.

Remark: for the base implementation one can get 2 points and implementing each enumerated feature provides an additional 1 point, but an explanation of the feature and showing explicitly the place of implementation are required.

[2p+5×1p]

2. Extend the repository created in the previous exercise with additional Find methods supporting filtering, paging and sorting, i.e.:

- for filtering *Find* takes additionally parameter (or parameters) which allow to apply some kind of filter for selected column (simple version) or columns (more mature version),
- for paging *Find* takes additionally *skip* (define starting record) and *take* (define how many records) parameters which allow to take specific subset of records from the result set;
- for sorting *Find* takes additionally which column (simple version) or columns (more mature version) should be used for sorting the results.

Propose a method (or methods) being a combination of all the above.

[3p]

Paweł Rajba