

Sport Equipment Portal

The program implements the back-end of a sport equipment evaluation portal.

The main class is Sports. The class TestApp in the package example contains an example. The exceptions are thrown using the class SportsException.

R1: Activities and Categories

The method `defineActivities(String... activities)` defines the sport activities supported by the portal. The method throws an exception if no activity is provided.

The method `getActivities()` returns the list of the defined activities, sorted alphabetically.

The method `addCategory(String name, String... linkedActivities)` adds a category of products and the activities related to that category. An exception is thrown if any of the activities has not been previously defined.

The method `countCategories()` returns the number of categories that have been defined.

The method `getCategoriesForActivity(String activity)` generates an alphabetically sorted list containing the category names related to the given sport activity. Returns an empty list if either no category has been defined for the activity or the activity does not exist.

R2: Products

The method `addProduct(String name, String activity, String category)` adds a product with its name, activity and the category it belongs to. The method throws an exception if a product with the same name already exists.

The method `getProductsForCategory(String categoryName)` returns a list of product names in the given category, sorted alphabetically. If the category has not been defined or does not contain any product, an empty list is returned.

The method `getProductsForActivity(String activityName)` returns a list of product names for the given activity, sorted alphabetically. If the activity has not been defined or there are no related products, an empty list is returned.

The method `getProducts(String activityName, String categoryNames)` returns a list of product names sorted alphabetically, that belong to the given activity and to one of the

given categories. If the activity is not defined or it has no products, an empty list is returned.

R3: Users and Ratings

The method `addRating(String productName, String userName, int numStars, String comment)` adds a rating for a product from a user of the system. A product is rated with a number of stars from 0 to 5 and a comment. An exception is thrown if the number of stars is not between 0 and 5 (included).

The method `getRatingsForProduct(String productName)` produces a list of ratings in the following format: "`# : [Comment]`", where `#` is the number of stars, while `[Comment]` is the associated comment. The list has to be ordered by the descending number of stars. An empty list is returned if no user has provided a rating for the given product.

R4: Evaluations

The method `getStarsOfProduct(String productName)` returns the average number of stars for a given product.

The method `averageStars()` calculates the average number of stars for all the ratings.

R5: Statistics

The method `starsPerActivity()` returns a map that associates a name of the activity to the average number of stars for the products belonging to that activity, with the activity names sorted alphabetically. Activities whose products have not been rated should not appear in the result.

The method `getProductsPerStars()` gives a map that associates the average number of stars with the list of product names that have this average, with the average values sorted in descending order and the names of the products sorted alphabetically. Products without reviews are discarded.