

PROO/SEW 3 - Assignment: Article Management

Objective

Create a program for managing a supermarket's articles.

Things To Learn

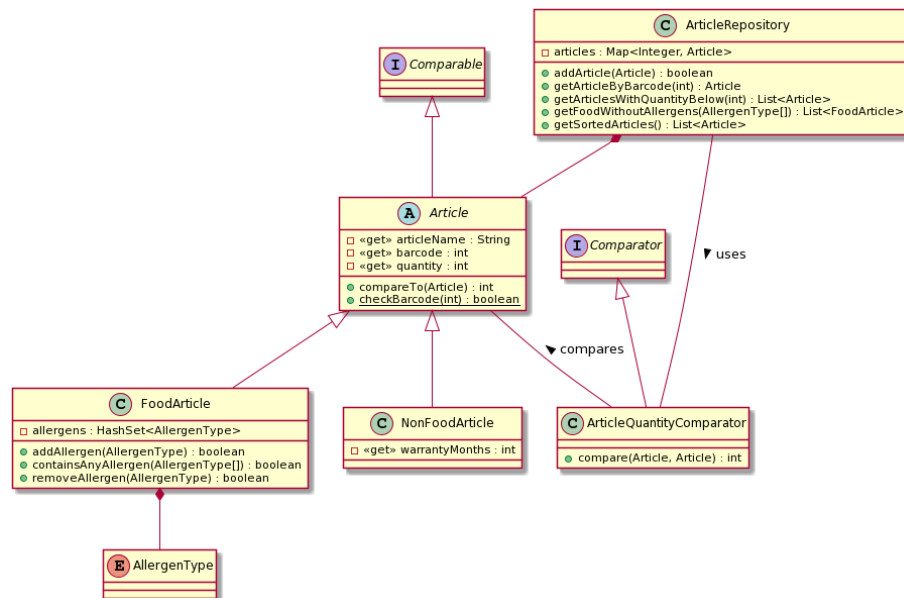
- Working with *hash maps*.
- Working with *enums*.

Submission Guidelines

- Your implemented solution as **zipped** *IntelliJ*-project.

Task

You are tasked with implementing the article management for the new supermarket *Billa Plus Plus* (or *B++* for short). Each article can either be food (including a list of allergens) or non-food, and can be uniquely identified by its barcode. Take a look at the following diagram and the implementation details below:



Article

- The **abstract** base class for all articles.

- Negative article quantities should be corrected to 0.
- Implements the `Comparable`-interface using the article name for ordering.
- Provides a `static`-method for checking the validity of a barcode.

***EAN-8* Validation**

The *EAN-8* barcode is used for small packages such as chewing gums or cigarettes - or in our case small data types like integers. This globally unique item number shares the same validation algorithm as its *big brothers* having 12, 13 and 14 digits.

The following diagram demonstrates the algorithm for the barcode *7351353X*:

Number	7	3	5	1	3	5	3
Weight	1	3	1	3	1	3	1
Products	7	9	5	3	3	15	3
Sum							45
Sum <i>mod</i> 10							5
Difference to next 10							5

Thus, the complete valid barcode is *73513535*.

FoodArticle

- The class for all groceries, inherits from `Article`.
- Additionally stores the food's allergens in a `HashSet` using the *enumerated type* `AllergenType`.
 - Currently there are 14 food ingredients that must be declared as allergens: A, B, C, D, E, F, G, H, L, M, N, O, P, R.
- The constructor accepts an array of allergens. The `HashSet`'s `addAll`-method could prove to be useful for this.
- The class provides some methods for managing a food item's allergens. Both `addAllergen` and `removeAllergen` should return `false` if the allergen can't be found, while `containsAnyAllergen` should return `true` if the food item has *any* of the passed allergens.

NonFoodArticle

- The class for all non-food items, inherits from `Article`.
- Negative warranty months should be corrected to 0.

ArticleRepository

- Stores the `Articles` in a *hash map* using their barcodes as keys.
- Provides various methods for adding and accessing articles:

- `addArticle` adds an article to the map and returns `true` if successful. Articles need to have a valid and unique barcode!
- `getArticleByBarcode` returns an article by the barcode passed.
- `getArticlesWithQuantityBelow` returns a list of articles with the stock below a passed value. The articles should be sorted by their quantity.
- `getFoodWithoutAllergens` returns a list of articles (sorted by their names) not containing **either** of the passed allergens.
- `getSortedArticles` returns a list of articles sorted by their names.

Additional Task

The `getArticleByBarcode`-method's behavior in the case of non-existing barcodes has not been defined - neither in the documentation above nor in the unit tests. Make the method throw a self-implemented exception and write appropriate unit tests.