Code documentation for Checkpoint_2

The task was to create at least two different classes, and to enter an error message in case of incorrect entry. And that you could also search for an article and mark the keyword in the search result itself.

I chose to go a few steps further in terms of challenging myself, but also to make it easier for the users of the program. Some attempts have been made to make the code a little more readable by breaking down large functions into a number of smaller sub-functions.

In other words, the task would be of the OOP type, I chose to go a step further with the files and chose to work according to the MVC methodology. To make it easier for the user, I also chose to build a flexible menu system for the service, and that the various categories and articles are saved locally to the computer. Had it been a program that would be used in reality, I would not have done it as a console application but as a website with a database connection, etc.… Now I chose to follow the intended syllabus!

Below are the various folders with a brief comment on what they contain and any methods / purpose.

Classes

Classes contain definitions for different classes used by the program, I have chosen to collect these in a separate catalog for the sake of simplicity but also to make it easier for the future when maintaining the code.

Articles

Class for handling different articles which are then added to a list, this also includes a couple of different constructions that are needed by the program, mainly when loading data from a file.

All variable names should be self-explanatory and what their purpose is!

Category

Class to handle the different categories used in the service, there are also some constructors needed to load data from file.

All variable names should be self-explanatory and what their purpose is!

Mastersettings

Class to specify certain parameters that should apply in general or when an article or category is added to the program. All variable names should be self-explanatory and what their purpose is!

Control

Control contains classes and functions for verifying syntax of mainly entered data and that this is in the program.

AddArticle

Is not a direct class to note, it has only one purpose and that is to add an article to our register. It starts with a couple of local variables that only exist when the function is called. VS whined that it could not always find certain local variables and realized that I had to solve it like this.

addProduct()

In parameter : None

Out parameter: None

Invokes functions to verify the syntax on which category the article should be linked to but also the actual syntax for article id. It also requests from the user a few different parameters needed by the program.

AddCattegory

addCategory ()

In parameter : None

Out parameter: None

Function for adding a new product group to our register, it makes sure to call functions to verify the syntax of the product group ID and also checks if it already exists. This also applies to the name! In a web-based store, the same category name would have been allowed to make it easier for visitors if we have e.g. a golf shop, we might want the manufacturer Ping to appear under many different product groups such as iron and hybrids etc.

ChangeCategoryProperties

changeCategeryProperties (string oldCategoryId = null, string newCategoryId = null, string newCategoryName = null, bool combineCategorys = false)

In parameters;

string oldCategoryId Old category id that we are going to change

string newCategoryId The new id't we want to change to

string newCategoryName The new category name we are going to put / keep bool combineCategorys Parameter for whether we should merge the existing category into one and the same and that all articles should be transferred to the new joint.

Out parameter: None

Function for editing different properties for a category, such as the name, category ID and the standard VAT rate.

CheckArticleSKU

checkArticleSKU (string searchSKU, bool existing = true)

In parameters:

string searchSKU, article SKU that we should search for.

bool existing, if we are to present that the article already exists for the user.

Out parameters: bool for if the article's SKU exists in the system.

Function to check whether an article SKU is in the system or not.

CheckCategoryId

checkCategoryId (string searchId, bool existing = true) In parameters:

A parameter;

string searchId, category id that we should search for.

bool existing, if we are to present that the category already exists for the user.

Out parameters:

bool for if category id exists in the system.

Function to check if a category ID is in the system or not.

CheckSyntax

A class with a number of common private variables and private functions to verify the syntax for different SKUs or category ids that exist.

ClassExist (string className)

A parameter

string className, the class name we are going to verify exists

Out parameter:

Bool, true if the class exists, otherwise false.

convertFieldsToDictonary ()

In parameter : None a parameter

Out parameter: No out parameter

Converts a fieldInfo variable type to a dictonary to facilitate further coding.

checkForceHyphen ()

In parameter : None a parameter

From parameter: bool

Function to verify if received string to verify contains the correct number of hyphens, etc. Returns true if everything is okay, otherwise false.

checkMinLengthLeftSide ()

In parameter : None a parameter

Ut bool

Function to verify the minimum length of category id or for the left part if we have an article number that consists of two different parts that are tied together with a hyphen.

check (string className = null, string InputString = null)

In parameters:

className classname that we work by and its associated parameters

InputString close as we check the syntax

From parameter: bool

Main function for verifying the syntax for a category id or article SKU that we want to verify. Returns true if all conditions are correct otherwise false.

FileRegister

Class file for writing and reading the different lists for categories and articles that we have in our register.

readRegister ()

In parameters: None in parameters

Out parameter: No out parameter

Function for loading the JSON code string contained in the two different files into the lists used by the program.

saveRegister

In parameters: None in parameters

Out parameter: No out parameter

Function for writing down the two different lists with categories and articles for the hard disk in JSON format in the form of text files.

Model

RemoveArticle

Class file to delete an article from our article directory.

removeArticle (string inputSKU)

A parameter:

String inputSKU, article number of the article we are going to delete from the register.

Out parameter: No out parameter

ModelSearchCategory

Class file to delete a category and, if desired, also linked products.

removeCategory (string inputCategoryId)

A parameter:

string inputCategoryId, which category id to delete from the category register located in the system

Out parameter: No out parameter

VATclass

Class to define the different VAT rates used here in Sweden

buildVatDictonary ()

In parameter : None a parameter

Out parameter: No out parameter

Class to define the different VAT rates used here in Sweden in a general VAT dictonary.

View

EditArticle

Some private variables are used that are used by the function in the class to begin with. From the beginning, it was an even larger function that was difficult to overview and was therefore divided into a number of smaller private functions.

showSelectArticleToEdit ()

In parameter : None a parameter

Out parameter: No out parameter

Function for examining whether the article you want to edit really wants to edit. Should probably be under control on closer inspection.

showSelectCategoryForProduct ()

In parameter : None a parameter

Out parameter: No out parameter

Function to specify which category you want the product to belong to. Can be discussed whether it should be under View or control.

showEditArticleName ()

In parameter : None a parameter

Out parameter: No out parameter

Function for editing the article name.

showEditArticlePrice ()

In parameter : None a parameter

Out parameter: No out parameter

Function for editing the price of an article.

showArticleEditVAT ()

In parameter : None a parameter

Out parameter: No out parameter

Function that calls a function to edit which VAT group the item should belong to.

saveEditToRegister ()

In parameter : None a parameter

Out parameter: No out parameter

Function to save any adjustments to an item. Also call the function to save articles and categories.

editArticle ()

In parameter : None a parameter

Out parameter: No out parameter

A public function for editing different properties of an article.

EditCategory

Class file for editing different properties for a category, e.g. Name and category ID but also the standard VAT rate for items included in that category. Tried to split a large function into smaller part functions to make the file more readable.

editVATCategory ()

In parameter : None a parameter

Out parameter: No out parameter

Function for editing which VAT rate should apply to a category or article. Saves the selection to a general variable which is then retrieved when a category or article is to be saved after editing.

showEditCategory ()

In parameter : None a parameter

Out parameter: No out parameter

Function for editing the name of a category.

showSelectCategory ()

In parameter : None a parameter

Out parameter: No out parameter

Function for selecting which product group you want to edit.

Menu

Class file to present a menu to the user, it is easy to add or remove options.

displayMenu ()

In parameter : None a parameter

Out parameter: No out parameter

Function that renders and calls selected functions!

RemoveArticle

removeArticle ()

In parameter : None a parameter

Out parameter: No out parameter

Function for deleting a specific item.

RemoveCategory

removeCategory ()

In parameter : None a parameter

Out parameter: No out parameter

Function for deleting a specific category.

Search

Class for doing searches within articles or categories

Search

In parameter : None a parameter

Out parameter: No out parameter

Function to decide if you want to search among articles or product groups.

searchArticleName ()

In parameter : None a parameter

Out parameter: No out parameter

Search function based on article names.

searchCategoryName ()

In parameter : None a parameter

Out parameter: No out parameter

Search function based on the names of the categories.

ShowArticles

Class to present the various articles in our article register.

showArticles ()

In parameter : None a parameter

Out parameter: No out parameter

Function for presenting the different articles based on different options such as price / product group / article.

ShowCategorys

A class to present the different product groups based on the desired sorting.

showCategorysById ()

In parameter : None a parameter

Out parameter: No out parameter

Function to present the categories based on its category id.

showCategorysByName ()

In parameter : None a parameter

Out parameter: No out parameter

Function for presenting the categories based on category names.

showCategorys ()

In parameter : None a parameter

Out parameter: No out parameter

Function that presents different options for the user about how it wants the result presents for the different product groups.

ShowCategoryVAT

Class to present the different VAT rates defined in the program.

showVAT (= int defaultVAT Mastersettings.defaultVATclass, bool updateProducts = true)

A parameter:

int defaultVAT which default VAT rate is to be presented to a product group if this is set. Otherwise, standard VAT is used, which is defined in general terms.

bool updateProducts if we are to also update the VAT group for all products that are linked to that particular product group.

Out parameter: No out parameter

Function for presenting and selecting VAT categories for product group or article

Root directory

Program

Begins by defining a number of lists and dictionaries as of the program.

Main()

In parameter : None a parameter

Out parameter: No out parameter

The main program itself, which starts up the entire program, calls some functions, among others. to load product groups and articles and call the presentation of the menu.

Anders Wallin, Helsingborg 2022-01-09