



# Home accounting

## The Programmer Documentation

Jakub Neumann  
(NEU0074)

## 1. Introduction

This document contains brief information about a program: it's code structure, principle of working and other. For information how to use the program, refer to User Documentation. For more detailed information, refer to Doxygen generated documentation.

## 2. The code structure overview

The code structure is as follows. In the beginning, right after including the libraries, we can find definition (and/or declaration) of global variables. Below, there are structure and table of categories (along with its size) defined and declared, as well as other variables used for storage of further counted values. There we can find declaration of functions, which are then defined below the main function.

The main function is designed to be short and rather call other functions (dealing with input file, counting the budget, sorting the records and creating the output file and writing it, as well as displaying the text to a console, zeroing the tables and converting integer type to string one) than perform them itself, so the code is clear to the programmer.

## 3. Operation algorithm

The first step is to open the input file. If the operation succeeds, the number of non-empty lines in the file is counted. This number is used for dynamic allocation of structure [budget], responsible for storing all the records. Then, the particular elements of structure (ID, type, category, amount and date) are filled with all the records.

All of the next steps are performed in a loop, for every month in a year, plus for total summary of the whole year.

For each possible month, it is checked if it is present in the dates in the input file. If yes, the month name is written to the output file and calculations allowed. Along with checking the presence, the number of records with a proper date is counted. Then, a local table of structures for storing only the records of this month is created and filled along with counting the amount of money for every category as well as income and expenditure. Next step is sorting the records in table by the amount of money (from biggest to lowest) with the use of bubble sort algorithm. Last part is writing the tables in .html file containing the results of above calculations: sorted records, sums of amounts of categories and summary of the month. The last two also apply to the year calculations.

## 4. Adjusting categories

The code is prepared for easy change of categories in the budget. To do so, it is just needed for the user to change the elements of the table that stores them (table [string names] defined in the beginning of main.cpp file). The categories' names might be changed, according to the user's preferences. Also, they might be removed, as well as added.

## **5. Additional information**

For the clearance of the code, variables' (as well as structures' and tables') names begin with lowercase letter and functions' names begin with uppercase letter.

All of the html markers are used only in functions that names of are "Write[...]".

The code is not errorproof if it comes to the input file that must be created accordingly to the rules described in The User Documentation.