Overview

The Coursework 2 is actually the continuation of Coursework 1. The main difference is that now the data structures are implemented using the C++ STL containers. The indeces of items (1…10) and data structures (1…5) are as you used in Coursework 1.

Initial data

There are 10 different classes of items (*ITEM1, ITEM2, …, ITEM10*). Their declarations are in file Items.h stored in Files for coursework #2.

DLL *DataProvider.dll* (also stored in Files for coursework #2) imports function *GetItem()* that constructs a stand-alone item and returns the pointer to it. It needs auxiliary file *Colors.txt*, created from https://en.m.wikipedia.org/wiki/Lists_of_colors. Declaration of *GetItem()* is in file *DateProvider.h*.

Task 1

Implement class *Item*:

```
class Item : public ITEMi
{
public:
        Item() { }
        Item(char *pID);
        ~Item();
        Item(const Item& Orig);
        Item& operator=(const Item& Right);
        bool operator==(const Item& Other);
        friend ostream &operator<<(ostream& ostr, const Item &it);
        char* GetID() { return pID; }
};
```

Text printed in blue depends on the type of your item (1…10), specify it yourself.

The constructor must call *GetItem()* from the DLL. Do not forget to relase the auxiliary memory fields.

Task 2

Implement class *DataStructure* containing the following members:

1. Depending on the number of your struct (1…5)[1]:
   map<char, map<char, forward_list<Item>>*> *pStruct;*
   or
   map<char, array<forward_list< Item>, 26>*> *pStruct;*
   or
   *array<map<char, forward_list< Item *>>, 26> pStruct;*

[1] This attribute must be private. The following functions must be public.

or
list<map<char, forward_list< Item >>*> *pStruct;*
or
list<array<forward_list< Item>, 26>*> *pStruct;*

2. *DataStructure();*
   Constructor that creates empty data structure.

3. *~DataStructure();*
   Destructor.

4. *int GetItemsNumber();*
   Returns the current number of items in data structure.

5. *Item *GetItem(char *pID);*
   Returns pointer to item with the specified ID. If the item was not found, returns 0.

6. *void operator+=(Item& item) throw(std::exception);*
   Operator function to add a new item into data structure. Fails if the data structure already contains an item with the specified ID. Usage example:
   *DataStructure *pds = new DataStructure;*
   Item it(nullptr);
   **pds += it;*

7. *void operator-=(char *pID) throw(std::exception);*
   Operator function to remove and destroy item with the specified ID. Fails if there is no item with the specified ID. Usage example:
   **pds-= buf; // array buf contains the ID*

8. *friend std::ostream &operator<<(std::ostream &ostr, const DataStructure &str);*
   Prints all the items into command prompt window. Usage example:
   *cout << *pds << endl << endl;*

Requirements[i]

1. In function *operator<<* you must use the range-based *for* loops, STL algorithm *for_each* and lambdas.

2. In function *GetItem* you must use the range-based *for* loops, STL algorithm *find_if* and lambdas..

3. Destructor must delete all the items.

Evaluation

The student's work is accepted if the evaluation test funcion runs correctly and produces all the supposed results. The evaluation test function is in file *Test.cpp* stored in Files for coursework #2.

The deadline is the week 16 of the semester (i.e. May 19). However, it is strongly advised to present the results of coursework earlier. The students can do it after each lecture.

Presenting the final release is not necessary. It is OK to demonstrate the work of application in debug mode of the Visual Studio environment.

To get the assessment the students must attend personally. Electronically (e-mail, GitHub, etc.) sent courseworks are neither accepted nor reviewed. The students may be asked to explain their code or even right on the spot write a small modification.

Read also section *Hindamine* from Ülevaade.

First steps

They are as in Coursework 1. Remember that instead of *DataSourse.dll* we use now *DataProvider.dll*. The only file to include is *Items.h.*

[i] Requirements to Coursework 1 are still valid