Accounting formulas implementation in C++ applications

Marius Popa
Melos AHMETI
Imer HALITI
Besim KAMBERAJ
MA Students BRIE

Abstract

In this paper are presented some concepts regarding the accounting, especially the financial accounting. Also, there are presented some examples on inventory activities and the way in which these are implemented in C++ applications, using various data structures. There are presented some considerations regarding the software using in accounting.

1. Introduction

Accounting helps management, shareholders, and banks, outside creditors to evaluate the entity performance, if the business is running well. Today this is known as the language of business. Anyone who wants to step in the world of commerce must understand accounting. Same as the bank oil the wheel of commerce, accounting does the wheel of investments, management and cash flow as well.

Generally we distinguish many types of accounting like:

- financial accounting;
- managerial accounting;
- government accounting;
- inventory accounting;
- tax accounting, used for the purpose of report to the authorities for year end tax return.

In this paper we are going to focus on the financial reports of a business entity as limited partnership. As the owners here appear two persons holding each 25% of capital and 50% belongs to a franchiser. This means by the end of year when the fiscal year ends a profit will be shared according to numbers stated above.

The company employs two different types of employees. One engaged for office and administration works and other one for research work on the field as the company does researches on variable socio-economic issues and trends. The first group is paid on the fix basis salary. For this group the company knows in advance what would be the costs for the following period.
The second group is paid on hourly basis, e.g. 2€ per working hour. It remains for the accounting department to calculate income taxes and contributions from the gross salaries.

Example: Calculate the income tax for the gross salary of the secretary after the pension contribution has been deducted.

Table 1 Income tax calculation

Name	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**	**A11**	**A12**	**A13**	**A14**
Marius Popa	600	30	570	80	0	170	8.5	200	20	120	24	52.5	517.5
Melos Ahmeti	500	25	475	80	0	170	8.5	200	20	25	5	33.5	441.5
Imer Haliti	400	20	380	80	0	170	8.5	130	13	0	0	21.5	358.5

where:

- A2 – Gross salary, amount reported to the authorities
- A3 – Pension Contribution (5% of the gross salary)

- A4 – Amount taxable after deduction of contributions
- A5 – Taxable amount for the first grade: 0€ – 80€
- A6 – Income tax for the amount 0€ – 80€ = 0€
- A7 – Taxable amount for the second grade: 81€ – 250€
- A8 – Income tax for the amount 81€ – 250€, 5% of the amount
- A9 – Taxable amount for the third grade: 251€ – 450€
- A10 – Income tax for the amount 251€ – 450€, 10% of the amount
- A11 – Taxable amount for the last grade: over 451€
- A12 – Income tax for the amount over 451€, 20% of the amount
- A13 – Total income tax payable to the authorities
- A14 – Net salary to be paid to employee

The implementation in C++ of indicators above is made with a record structure. A record is a collection of data, being an aggregate data type. In a record the data items can be of different types, in contrast to arrays. In C++ the record structure is called structure and the keyword is *struct*.

To access the data item in a record structure it is used the operator dot “.”. The record items are local to the structure and they are known outside the structure.

The record structures used in the C++ program that implements the example above are:

```
struct taxable
{
	float grade;
	float i_tax;
	float val_tax;
};
struct fields
{
	float g_sal;
	float p_contrib;
	float amount;
	taxable income[4];
	float total;
	float net_sal;
};
```

The structure *taxable* is included in *fields* structure. The structure *field* hold the values of the indicators above.

The class *tax* has the following structure:

```
class tax
{
public:
	fields ex;

	tax();

	void p_contrib_val();
	void amount();
	void total_tax();
	void net_salary();

};
```

The methods are:

- *p_contrib_val( )* – computes pension contribution value;
- *amount( )* – computes amount taxable after deduction of contributions;
- *total_tax( )* – total income tax payable;
- *net_salary( )* – net salary to be paid.

The C++ application returns the value of total income tax payable and net salary. This value depends of the grades and income taxes established by user

2. Inventory accounting

The sales industry differs from other. Also the accounting for sales companies differs on that depending from the volume of sales, inventories and working personnel. We distinguish two types of track-keeping. The first one, perpetual method, is popular among industrialized countries and where the gross sales and retailers are in steps forward.

Periodic inventory system has following characteristics:

- does not keep track of inventory, except periodically;
- cannot distinguish between goods sold and goods stolen, lost, or destroyed and is calculated with formula:

Beginning Inventory + Purchases – Ending Inventory = Cost of Good Sold

For example, in the store of a small retailer are employed two persons: salesman and cashier, therefore there is no time for a detail accounting record after each transaction has been done.

We assume followings:

- beginning inventory as of 01.01.2002 amounts 20,000 €
- purchases during the year: 100,000 €
- ending inventory as of 31.12.2003 amounts 44,000 €

Using the numbers above we calculate Cost of goods sold:

20,000 + 100,000 – 44,000 = 76,000 €

The scheme below shows us the procedure:

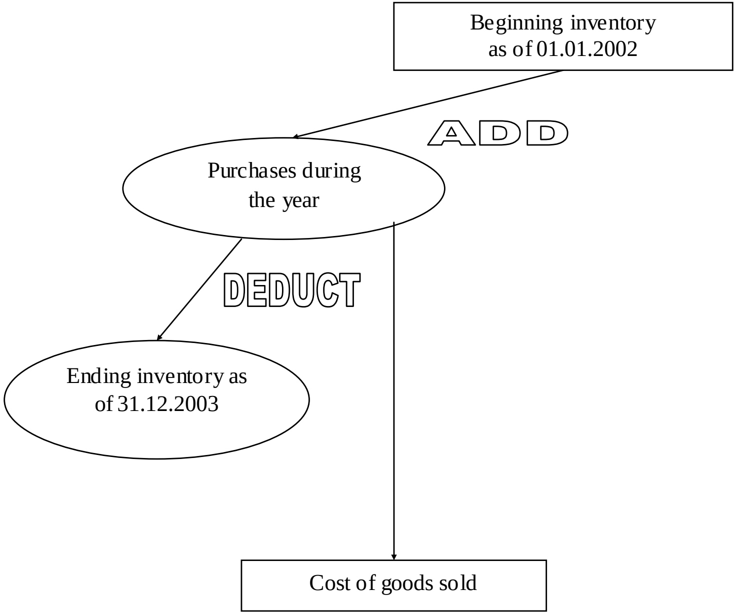
Beginning inventory as of 01.01.2002
ADD
Purchases during the year
DEDUCT
Ending inventory as of 31.12.2003
Cost of goods sold

Figure 1 Cost of Goods sold calculate procedure

Transactions:

Cash or Account Receivable = Sales   150,000.00 €

Cost of goods sold = Inventory   76,000 €

Record of the sales and cost of goods sold during the period 01.01.2002- 31.12.2002.

Income Statement for 2002:

- revenues: 90,000 €
- expenses: 76,000 €

Income/Loss before taxes:

Revenues – Expenses = 14,000 €

Upon each sale, debit cash, or accounts receivable, and credit revenue.

In contrast, a perpetual system would also debit cost of goods sold and credit inventory, resulting in a four part entry.

The example above was implemented in C++ programming language by a simple linked list.

A single is defined the same way as any other? User defined type or object, except that it also contains a pointer to the variable of a same type as itself.

A linked list is a structure containing a list of objects. One object is denoted as the first. Each object links to the next one in the links.

Linked list can be open or closed, if they are closed, the last object links back round to the first, and a vice versa to the double linked list.

The C++ application implements the following operations, the nodes being accounts:

- adds a new node in a list;
- deletes a node in a list;
- finds a node in a list with specific information;
- computes the revenues;
- computes the expenses;
- computes income/loss before taxes.

The structure of a node is:

```
struct account
{
	int no;
	char name[30];
	char type;
	float begin[2];
	float purchase[2];
	float end[2];
};
struct node
{
	account acc;
	node *next;
};
```

Perpetual inventory system has the following characteristics:

- requires more record keeping than does a periodic system.
- more common recently because of scanner devices.
- provides continual information regarding what was sold, where and when it was sold, and at what price.

- a periodic inventory count – typically annually – is still necessary**.**
- till requires a choice of valuation methods

Below there are presented cost of goods sold methods of valuation.

LIFO (Last In First Out). According with this method of valuation, the last inventory purchased was the first sold and the original inventory purchased remains.

FIFO (First In First Out). According with this method, the first inventory purchased was the first sold and the last inventory purchased remains.

Average cost. Values inventory sold at the average cost of inventory purchased.

Specific identification method. This method of valuation keeps track of each item and its own specific cost. It is more useful for very large or valuable items which are not fungible such as for airline manufacturing.

3. Conclusions

We have tried to stress some aspects and applicable methods/valuations on today's business computations, transactions and communications using accounting tools among different units within economic system itself. The strong flow of information, the design of informatics system depends mainly on that how the accounting system functions, how the data structures are used in accounting, the level of education and professionalism among working personnel and support of software packages as well.

The importance of data structures used in accounting has no alternative. This is described on the chapters above and explained relationships between them. On the first part we focused on profit share among owners and shareholders of company and how the salaries are calculated on progressive taxation depending on different tax grades and amount of gross salaries.

Also we focused in very important track-keeping and inventory accounting which is usable almost in every small, medium or large business. We recognized that the company may choose whether to use perpetual inventory which requires strong software background or periodic inventory that is more capable for small or medium size businesses.

Bibliography:

[CRIS99] Valentin CRISTEA, Irina ATHANASIU, Eugenia KALISZ, Valeriu IORGA – Tehnici de programare, Editura Teora, Bucureşti, 1999

[SMEU01] Ion SMEUREANU, Marian DÂRDALĂ – Programarea în limbajul C/C++, Editura Cison, Bucureşti, 2001

[SMEU02] Ion SMEUREANU, Marian DÂRDALĂ – Programarea orientată obiect în limbajul C++, Editura Cison, Bucureşti, 2002

[www1] www.Inventory Accounting.htm

[www2] www.cs.wright.edu/~kpraveen/cs241/PPTSlides/Structures.pdf

[www3] www.sks99.com/ch9/mops.htm