ADVANTAGES OF USING OBJECT-ORIENTED TECHNOLOGIES IN MODELING COSTS

Assoc. Prof. Ph.D Mehedintu Anca
Prof. Ph.D Pirvu Cerasela
Prof. Ph.D Soava Georgeta
University of Craiova
Faculty of Economics and Business Administration
Craiova, Romania

Abstract: The project was created with the intention of helping the managers, whose objective is to optimize the use of resources so that they obtain the wanted profit. In the first paragraphs we presented the theoretical concepts that we had in order to make this application. We pointed out the necessity to pass to object oriented programming, underlining the main advantages that made us chose this type of programming. Next, we showed the importance of the production cost in the decisional process and its calculus methods.

JEL classification: M15, M41

Key words: object oriented technologies, costs management, optimization, linear programming, object -oriented programming

Introduction

Starting from the importance and the role of information regarding costs in decisional process, the purpose of this paper is to present the main tendencies registered in the evolution of the informational technologies to realize support informatics systems for assistance of the managerial decision regarding costs.

Modeling instruments are an essential part of the managerial process, taking into account the fact that modeling supposes the abstracting of the problem and its generalization in quantitative and / or qualitative forms.

Hereby, for the present paper we have used modeling by linear programming and we have realized an information application being based on the object oriented technology to answer the problems of costs optimization. We have elected this informatics optimization instrument that generalizes the technique of value as purpose and offers several possibilities of simulating some parameters giving birth to optimum situating.

The purpose of this application is the one to help the decisional process, being a beneficial support of the managers.

1. Object oriented programming

1.1. Principles that are at the basis of object oriented programming

Each programming model imposes a certain programming style that is in connection with the main concepts that characterize that model. The principles that are at the basis of object oriented programming are:

A) Abstracting – helps us delimit in a strict way "WHAT does the object" of "HOW does the object what it does".

Abstracting is one of the main modalities through which we, the people, manage to understand and enclose complexity, offering the possibility that a program to ignore some aspects of the information that it handles, namely the capacity to focus on the essential. Each object in the system has the role of an abstract "actor", which can devolve actions, can modify and communicate the status and can be in connection with other objects from the system not presenting the modality in which these facilities have been implemented. The processes, functions or methods can also be abstract, and then, are necessary some techniques in order to extend abstracting.

The attitude of an object is characterized by a multitude of services or resources that they offer to other objects. Such a behavior, in which objects, named server, offer services to other objects, named customers, is described by the so-called model customer- server, that we have taken into consideration at the realization of the application.

B) Encapsulation – named also hiding data, is the complementary concept of abstracting.

If the result of the abstracting operation for a certain object is the identification of its protocol, then encapsulation has to do with the selection of an implementation and treating it as a secret of that abstraction. The encapsulation ensures the fact that objects are not allowed to change the internal status of other objects in a different manner (but only through methods that are offered by that object); only methods of the objects can access its status.

Encapsulation is the process of elements subdivision which form the structure and behavior of an abstraction, being useful for the separation of the "contractual" interface from its implementation. From this definition it results than an object is formed of two different parts: **interface (protocol)** namely **implementation of this interface**. Abstracting is the process through which is defined the interface of the object; meanwhile encapsulation defines the representation (structure) of the object, together with the implementation of the interface.

Encapsulation offers independence between programs, operations and information. An immediate advantage is the one that different programs of applications can be divided on the same objects not knowing the entrance – exit mechanism (import/ export). The principle of encapsulation can be sometimes temporary, allowing a more or less free access to the information from an object, making possible the stability, in time, of electing data implementation.

C) Modularity is the property of a system that was decomposed in a set of cohesive and weakly coupled modules.

The classes and objects obtained as a result of abstracting and encapsulation must be grouped and afterwards deposited in a physical form, named module. Modules can be regarded as physical recipients where we announce the classes and objects resulted according to designing at a logic level. So, the modules form the physical architecture of the program.

Modularization resides in separation of the program in a number of modules that can be separately put together, if are connected (coupled) between. Languages bearing the module concept make, in the same time the distinction between the interface of the module and its implementation.

D) Classification – represents an arrangement of the abstractions.

The most important hierarchy of classes, in the object paradigm is: classes' hierarchy (relationship of "is a" type) and objects hierarchy (relationship of "part of" type).

- **Polymorphism** – is the ability to process different objects according to their type or class, to redefine models for derivate classes.
- **Inheritance (classes' hierarchy)** – organizes and facilitates polymorphism and encapsulation allowing the definition and realization of some specialized classes starting from class (general) which are already defined – these can share (and extend) their behavior not being necessary the redefinition of the same behavior. The inheritance defines a relation between classes in which one class shares the structure and behavior defined in one or more classes (according to the case we speak about simple or multiple inheritance)
- **Aggregation (hierarchy on objects) –** the relationship between two objects in which one of the objects belongs to the other object. The aggregation gives appurtenance of an object to another object.

1.2. Advantages of object oriented programming

In order to realize the application regarding the managements of costs, we have used object oriented programming due to advantages that this offers us: shaping the objects of the applications, modularity, re-utilization and extensibility of the code leading to a greater productivity and the development of a great quality of applications (Negrescu, 2007):

- The facility of designing and reusing the code, once the correctness of some objects' functioning in an application, is tested, these can be used without any problem in other applications, this advantage can be capitalized by realizing a library of objects.
- Abstracting allows designers to obtain an ensemble image, aiming the behavior of the objects and interactions between them, the details being buried in the composition of the objects.
- Data safety is ensured by the ability of the objects to behave like some "black boxes", to be used without recognizing their composition, which ensure the diminution of the frequency of apparitions and the effect of the errors connected to wrong usage of data types.

Shaping complex reality, object oriented techniques emphasize on data behavior, encapsulating in the object concept, data as well as possible operation on them. Object oriented programming is a good method of organization of programmers, POO's properties leading to a main compact and elegant code. Objects can better describe concepts that they represent, begin more logic and intuitive that the traditional module, with simple data structures. Besides the simplification advantages, separation brings a plus of flexibility for implementers, as several implementations can be used as the same interface. Implementations can be different in which concerns efficiency of time, space, price or quality of the documentation that is at their disposal, or any non-functional characteristics. Also, a unique implementation can satisfy several interfaces, in this case the implementation contains a union of required methods by any of the interfaces.

2. Management of production costs

2.1. Types of models for cost optimization

Starting from the fact that the central objective of the enterprise is the optimization of the resources consumption, so the production cost must be minimal to obtain a maximum profitableness, as follows, we have attempted to present several mathematic models of optimization.

The economic theory and practice determined the necessity of issuing some extremely varied models, so *according to the mathematical nature of the relations describing the connections of the systems,* the models can be linear and non-linear.

Linear models represent the simplest modality of approximation of the economic relations, due to the fact that restrictions, as well as functions are of first degree. Most frequently used sample aims linear programming, because, as part of models that is solves, can be found also the ones regarding costs, aspect to which we grant adequate attention.

Non – linear models have as a characteristic the fact that restrictions, as well as object function have a superior degree. For example, square models in which restrictions or the objective function are of 2^{nd} degree.

Depending on the modality in which the time factor is or is not taken into account, there are static and dynamic models.

Static models are based on functions whose parameters are independent of the time factor. Decisions taken based on solutions offered by these models are valid only for a short period, but able to be ulterior updated.

Dynamic models are mostly used for most of the economic phenomena, better approximating reality. These are characterized by time functions.

Costs management shapes and registers the economic fluxes, but these operations do not have a sense but to the extent in which are addressed to the managers and are used by them in the process of taking decisions. Once the cost fluxes are identified, it must be disposed an analyze net of the expenses that enter in their structure, describing at the same moment the relations in which are positioned with attended finalities.

2.2. Modeling costs by linear programming

Linear programming contains those methods to solve problems whose model is made of a system of linear restrictions and having a function that is still objectively linear. The problem of linear programming is presented under the form of a system of linear equations characterizing an economic process or phenomenon.

In this paper, we have proposed that this type of modeling to be developed from the informatics point of view for cost's management.

To better understand this phenomenon, we will consider as an example in which we suppose that a company produces **m** products P_1, P_2,..P_m, for which are used **n** resources R_1, R_2,..R_n (raw materials, materials, fuel, energy, work force, etc). The production of a unit from products P_i supposes an a_{ij} consumption from resource R_j ($i = \overline{1,m}, j = \overline{1,n}$). These consumptions "aij" are known in the specialty literature as technical coefficients. Prices for a unit from resource R_j are noted P_{rj} ($j = \overline{1,n}$). The enterprise planned to produce from each product the quantities q_i ($i = \overline{1,m}$), according to the order received (Pirvu, 2008).

The presented information can be systemized in a chart having the form (Table 1):

It is discussed the problem that, using the matrix technique, the managers of the enterprise can express in a matrix manner the following elements:

- technical coefficients (with line products and resources in columns);
- price of consumed resources
- unit cost of the products
- total consumptions for the launched order (for quantities planned to the produced);
- total cost for each product from the launched order;
- global analytical cost.

Table no 1

P/R	R_1 R_2 ..... R_j ..... R_n	Quantities scheduled under order
P_1	a_{11} a_{12} ..... a_{1j} ..... a_{1n}	q_1
P_2	a_{21} a_{22} ..... a_{2j} ..... a_{2n}	q_2
.	.	.
.	.	.
P_i	a_{i1} a_{i2} ..... a_{ij} ..... a_{in}	q_i
.	.	.
P_m	a_{m1} a_{m2} ..... a_{mj} ..... a_{mn}	q_m
Prices of resources R_j	p_{r1} p_{r2} ..... p_{rj} ..... p_{rn}	

The following matrices would be used:

- matrix of technical coefficients

Table no 2:

	R_1 R_2 ..... R_j ..... R_n	
A=	a_{11} a_{12} ..... a_{1j} ..... a_{1n}	P_1
	a_{21} a_{22} ..... a_{2j} ..... a_{2n}	P_2
	.	.
	.	.
	a_{i1} a_{i2} ..... a_{ij} ..... a_{in}	P_i
	.	.
	a_{m1} a_{m2} ..... a_{mj} ..... a_{mn}	P_m

- vector (column matrix) of the unit prices for every resource: $p = \begin{bmatrix} p_{r1} \\ p_{r2} \\ : \\ p_{rrn} \end{bmatrix}$

With the aid of these two matrices, it can be calculated the vector of unit cost of the products P_m. We note:

$$C_u = \begin{bmatrix} C_{u1} \\ C_{u2} \\ : \\ C_{um} \end{bmatrix}; \text{ then } \quad C_u = \begin{bmatrix} C_{u1} \\ C_{u2} \\ : \\ C_{um} \end{bmatrix} = AxP = \begin{bmatrix} a_{11} & a_{12} & .. & a_{1n} \\ a_{21} & a_{22} & .. & a_{2n} \\ .. & .. & .. & .. \\ a_{m1} & a_{m2} & .. & a_{mn} \end{bmatrix} x \begin{bmatrix} p_{r1} \\ p_{r2} \\ : \\ p_{rn} \end{bmatrix}$$

- the matrix of the quantities contained in the order (quantities planned to be produced for every product P_i): $Q = \begin{bmatrix} q_1 & q_2 & .. & q_m \end{bmatrix}$

With the aid of this matrix and of the matrix of technical coefficients, we can express the matrix of consumptions of resources for execution of the order. Marking this matrix M_r we have:

$$M_r = \begin{bmatrix} m_{r1} \\ m_{r2} \\ : \\ m_{rm} \end{bmatrix}^t \text{; then } \quad M_r = \begin{bmatrix} m_{r1} \\ m_{r2} \\ : \\ m_{rrn} \end{bmatrix}^t = QxA = \begin{bmatrix} q_1 & q_2 & .. & q_m \end{bmatrix} x \begin{bmatrix} a_{11} & a_{12} & .. & a_{1n} \\ a_{21} & a_{22} & .. & a_{2n} \\ .. & .. & .. & .. \\ a_{m1} & a_{m2} & .. & a_{mn} \end{bmatrix}$$

Multiplying this matrix with the matrix of the unit costs we obtain the matrix of the total costs for each product in the order.

We note:

$C_p = \begin{bmatrix} C_{p1} \\ C_{p2} \\ : \\ C_{pm} \end{bmatrix}$ matrix of the total costs that are afferent to each product. Then:

$$C_p = \begin{bmatrix} C_{u1} x\, q_1 \\ C_{u2}\, x\, q_2 \\ : \\ C_{um}\, x\, q_m \end{bmatrix}.$$

Knowing the total cost of each product, we can determine the total cost of the order that we mark with C_{tc}. So:

$$C_{tc} = \begin{bmatrix} 1 & 1 & 1 ... & 1 \end{bmatrix} x \begin{bmatrix} C_{p1} \\ C_{p2} \\ : \\ C_{pm} \end{bmatrix}, \text{ sau } C_{tc} = \begin{bmatrix} 1 & 1 & 1 ... & 1 \end{bmatrix} x C_p .$$

We reach the same result if we multiply the unit prices of the resources with the quantities of resources that are used for each order, namely the product of matrices: $P(Q*A)=C_{tc}$.

Considering that commercial society that produces separately products, and supposing that in this scale of products are included components C_1, C_2,.. C_n, that are delivered to customers L_1,L_2,..L_m, $m \geq 1$. Prices of each component are p_1, p_2, ... p_n with the aid of which , we can write the matrix of prices: $P = \begin{bmatrix} p_1 \\ p_2 \\ : \\ p_n \end{bmatrix}$

We propose ourselves to determine the turnover of the society, for this we first calculate the turnover for each customer L_i, ($i=\overline{1,m}$).

Noting:

q_{Lij}= number of C_j components that would be delivered to the customer L_i;

$q_{Li}=[q_{Li1} \quad q_{Li2} \quad ... \quad q_{Lin}]$;
CA_{Li}= turnover for customer L_i;
We have: $CA_{Li}= q_{Li}xP$

Namely: $CA_{Li}=[q_{Li1} \quad q_{Li2} \quad ... \quad q_{Lin}]x \begin{bmatrix} p_1 \\ p_2 \\ : \\ p_n \end{bmatrix}$.

The turnover for each customer can be written in a matrix form, this way:

$$CA_L = \begin{bmatrix} q_{L11} & q_{L12} & .. & q_{L1n} \\ q_{L21} & q_{L22} & .. & q_{L2n} \\ .. & .. & .. & .. \\ q_{Lm1} & q_{Lm2} & .. & q_{Lmn} \end{bmatrix} x \begin{bmatrix} p_1 \\ p_2 \\ : \\ p_n \end{bmatrix} = \begin{bmatrix} CA_{L1} \\ CA_{L2} \\ : \\ CA_{Lm} \end{bmatrix}$$

Noting $q_L = (q_{Lij})_{\substack{j=\overline{1,n} \\ i=\overline{1,m}}}$. Then $CA_L=q_LxP$.

The total turnover would be: $CA = \sum_{i=1}^{m} CA_{Li}$.

Matriceal, the above expression can be written:

$$CA = [CA_{L1} \quad CA_{L2} \quad .. \quad CA_{Lm}]x \begin{bmatrix} 1 \\ 1 \\ : \\ 1 \end{bmatrix}.$$

3. Case study – informatics application regarding cost's management

3.1. Information technology used in realizing the application – C#

The mathematical model being already created passes to the realization of the informatics application regarding cost's management that we have realized in C#.

We have elected language C#, as it is a modern language, object oriented, allowing programmers to build in a fast manner a large variety of applications for the Microsoft.Net platform, platform ensuring instruments and services for exploitation of the informatics systems and communications. It combines the productivity of the languages of fast development of the applications having a gross power of the language C++. The C# code is put together as a basis code, meaning that takes the advantages of the CLR services, including the interoperability of languages, garbage collection, increased security and support of the enriched versions.

Due to the object oriented elegant design, C# is a good choice for conceiving varied components – from applications at system level and up to realizing commercial applications at high level. Using simple concepts of C# language, these components can be easily transformed into Web XML services, allowing their accessing by the Internet by any program ruling on any operation system. C# has been designed in parallel with the .NET platform, realized by Microsoft and, that is why, shares with this a great number of characteristics. In this why are eliminated a series of software defects connected to non-assignment of the memory – in C++ might appear cases in which memory is not unassigned and is lost up to stopping the program or cases in which the memory is unassigned by a component but other component attempts ulterior to writes at this address as it has a pointer that believes as being valid (Smeureanu, 2004).

Advantages offered by C# synthesized this way:

Rapid Application Development – one of the most important objectives of Microsoft for C# is a support for RAD (Rapid Application Development).

Inter - platform functioning – internet languages must support, by definition, a inter- platforms functioning, so services function on a large scale of hardware and software. Much more, client oriented software can roll on several types of devices, including PDA and mobile phones.

Access to the characteristic resources of the platform – programmers requires most often access to the specific resources to write powerful target applications.

Platform independent functioning, C# compilator generated a flux of binary code which is interpreted by .Net runtime.

Access of application at platform specific resources, offers programmatic access to specific resources of the platform. A C# program can use, for example, Windows API to roll as a complete application under Windows 2000. The same program can roll on a PDA using the subset Windows CE API.

3.2. Presentation of the application "Management of the costs"

The application „Management of costs" has a friendly interface, attempting to help managers that do not necessarily have preparation in informatics. At the opening of the application, on the screen appears a connection window (figure 1).

Figure no. 1

At the moment of introducing the user and the password, we have access at the main window of the application (figure 2). As we observe here, we have 3 options:

Figure no. 2

- The main menu, offering us the opportunity of the access to information that are necessary for the realization of the proposed objective;
- Help, which allows the access at information regarding the usage modality for any of the options presented in the main menu;
- Exit – allows the exit from the application.

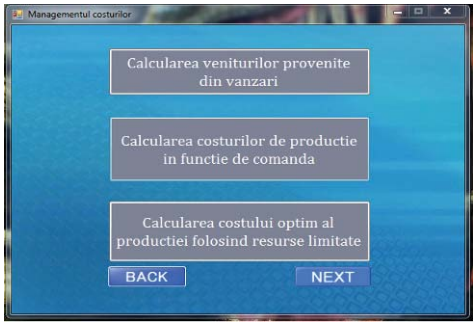


Figure no. 3. Components of the main menu

At the appeal to the first option, namely, the main Menu appears window from figure 3, having 3 main key buttons: Calculus of the turnover, Calculation of production costs depending on order and Calculus of the optimum cost of production, using limited resources, as well as 3 main key buttons that can be found in all windows: Back and Next offering the

possibility to return at the previous window or the movement to the following.

To better understand the way in which the application functions, we have taken as an example a commercial society with production of textile materials and products for tailory. We supposed that in a series of products are included several components that are delivered to one or several customers. So, selecting the key button for calculus of the incomes from sale it will appear a window in which we have to fill the number of customers, each of them ordering a number of products, and how many types of products are in the process of fabrication (figure 4).

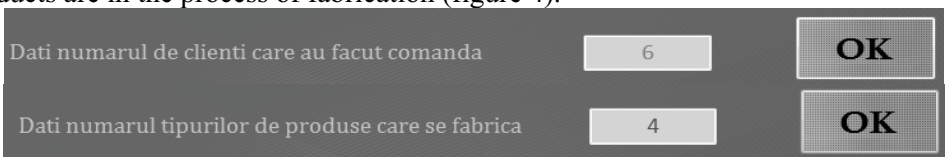


Figure no. 4

After the introduction of these information we have the opportunity to verify the correctness of data, while selecting the OK option, it will appear the window from figure 5.

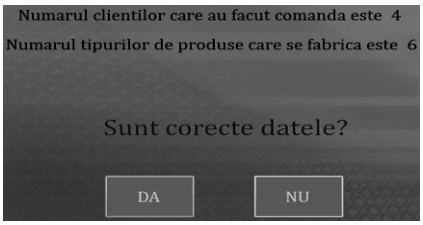


Figure 5

In case information have not been introduced in a correct manner and is pressed key NU, a new window will appear, in which the user would be able to modify either the number of customers or the types of products.

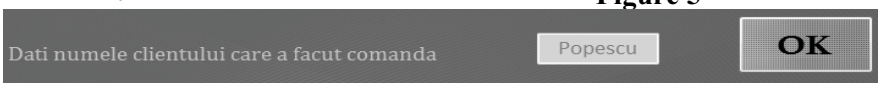


Figure no. 6

After the information have been successfully introduced, key DA is pressed (data are correct), in this moment we have the opportunity to introduce the name of each customer, in a list (figure 6), as well as the name of each product.

The user, after pressing key OK, it must be introduced the price of each product, separately (figure 7).

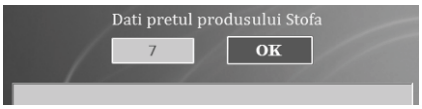


Figure no. 7

Also, in case data have not been correctly introduced, there is the opportunity to correct them by pressing key NU.

After the introduction of all entrance data, we would pass to the construction of the matrix of ordered quantities (figure 8). At each data introduction, it would be realized a verification, as it is mentioned in figure 9.

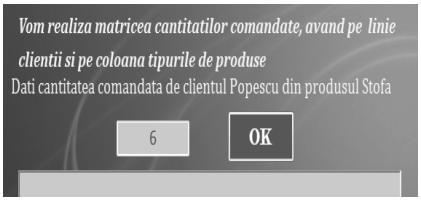


Figure no. 8

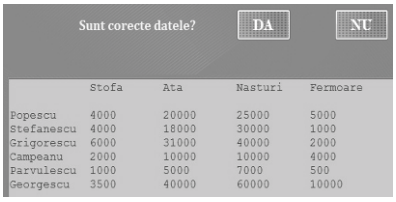


Figure no. 9

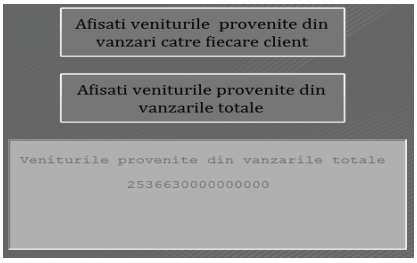


Figure no. 10

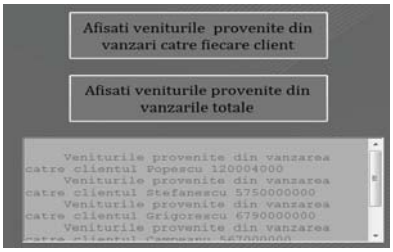


Figure no. 11

On the last formulary of the application can be found two keys, through the agency of which we can see reports for the calculus of incomes from sales (figure 10).

Also as part of this application of "Management of costs" can be calculated also the production costs, depending on the order by selecting the second option of the main menu.

To emphasis the generalized character of this application, we have elected to exemplify a commercial society producing furniture.

While selecting the proper key, there appears a window where would be introduced entrance data (number of the types of products that are produced and the number of used resources) according to figure 11.

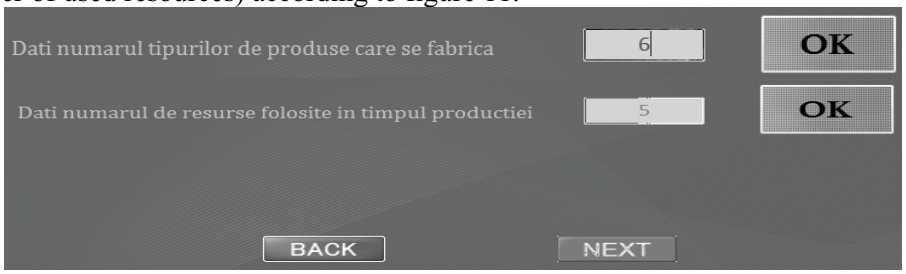


Figure no. 12

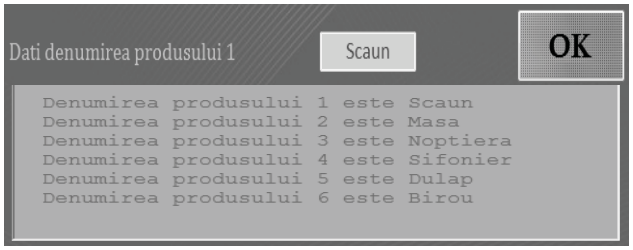
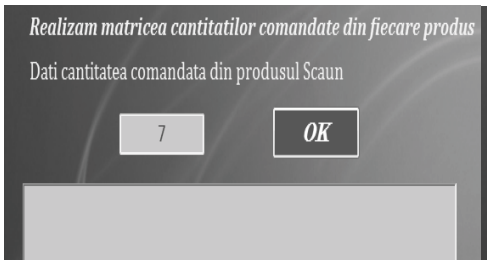


Figure no. 13

Similar to the example for the first problem, can be given names of the products (figure 12), obtaining a list with these products, after pressing OK key.

Ulterior, after being successfully quoted the name of the products as well as of the resources, the user has the possibility to introduce the cost for every separate resource, and



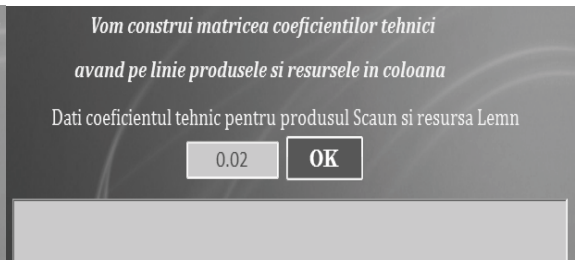
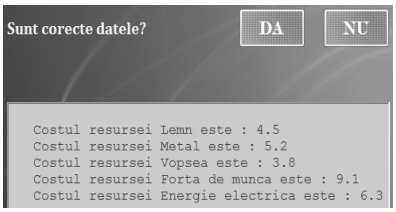


Figure no. 14 **Figure no. 15**

by pressing OK, these would be registered as it is shown in figure 13. As we have mentioned, it is validated the correctness of the information (figure 14, figure 15).



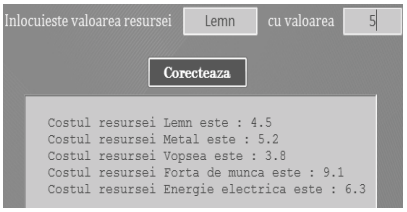


Figure no. 16 **Figure no. 17**

As follows, it is created the matrix of the ordered quantities and the one of technical coefficients (figure 16, figure 17).

After realizing matrices and data validation, there appears the last formulary of the application on which can be found four key buttons, with the aid of which we can visualize reports for calculus of costs: unit costs for each type of product, consumption from each type of resource, total cost of the order for each type of product and the total cost of the order (figure 18).

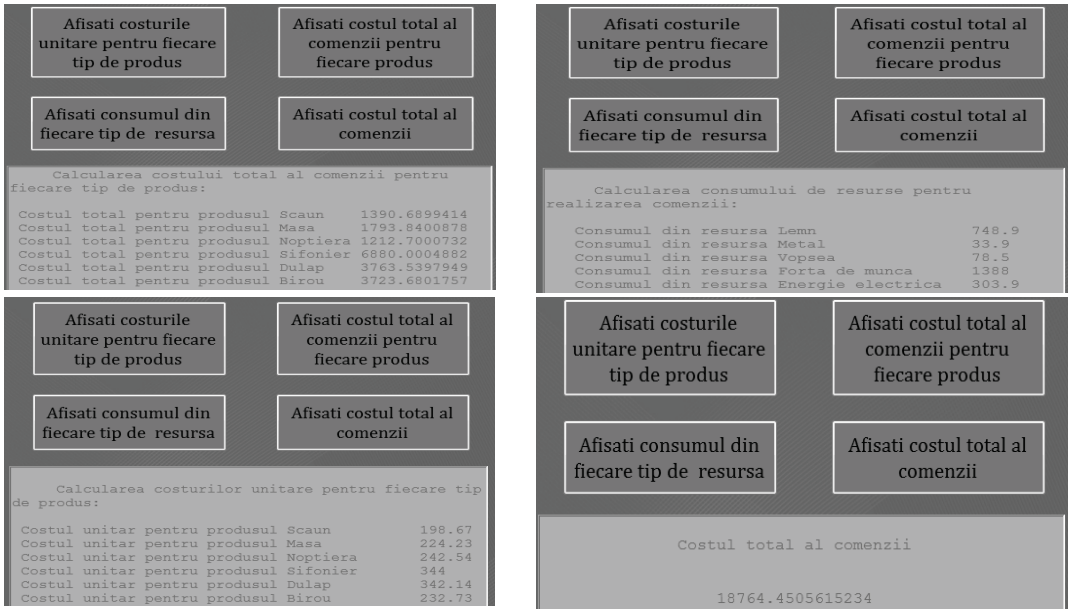


Figure no. 18

Conclusions

- In speed century, managers must take decisions as fast as possible, and in the decisional process, costs management occupies a very important place, which imposes a informational system, well adjusted.
- By this paper, we want to help managers, realizing an application calculating costs and incomes from production. For this, we used object oriented programming, due to easy design, the possibility of using the code again, abstracting and data safety.
- Thanks to constant progresses regarding languages and logics, it become possible the development of more and more complex informatics systems, so object oriented systems obtain a sum of the advantages of the management systems of data basis, interrogating in an efficient manner procedural data and languages, which allow complex calculus.
- Object orientation appeared as a solution to realize systems, complex applications, especially of stable elements of the system to uniform modeling on objects, underlining basis of the evolution of the applications and forecasts easy solutions of re-utilization in the evolution of logics. Modeling instruments represent an essential part of the managerial process taking into account the fact that modeling supposes conceptualization of the problem and its abstracting in quantitative and/ or qualitative forms.
- Management of costs models and registers economic fluxes, but these operations do not have sense but to the extent in which they address to managers and are used by them in the decisional process. Once the costs fluxes have been identified, it must be disposed an analyze net of expenses that enter in their structure, describing at the same time relations in which they are with the attended purposes.
- The main objective of the enterprise is the optimization of the cost of resources, o that the production cost be minimal to obtain a maximum profitableness, we have presented some mathematical models of optimization, stopping at the modeling of the linear programming that we have realized in this paper, from informatics point of view of costs management.
- We have elected C# language, as it is a modern language, object oriented, easy to be used, the internet offering much more help for this language, in comparison with the other, as it is used by specialists as well as by many amateur programmers.
- We consider that, once improved, the program that is realized can be integrated in an ERP system and put at the disposal of the users in the purpose of the management of business processes.

REFERENCES

1. Conger, D. ”Programming in C#”, Publishing House ALL, Bucharest, 2007
2. Giurca Vasilescu, L. ”Finanţele firmei. Prelegeri şi studii de caz”, Universitaria Publishing House, Craiova, 2009
3. Grenier, C. ”Systémes d’information et comptabilité, Encyclopédie de Comptabilité, Contrôle de Gestion et Audit”, Economica, Paris, 2000

4. Hannon, N.J. "The cost accountant is dead; long live the business process analyst", Strategic Finance, vol.87, no.6, 2005
5. Knuth, D.E. "Art of computer programming", vol.1, vol.3, Teora Publishing House, Bucharest, 2002
6. Negrescu, L. "C# language for beginners", volume I, II, III, IV, Albastra Publishing House, Bucharest, 2007
7. Pirvu, C. "Cost's calculus", Universitaria Publishing House, Craiova, 2008
8. Porter, M. "Competitive advantage – survival manual and of increasing companies in the conditions of the market economy", Teora Publishing House, Bucharest, 2001
9. Porter, M. "About competition", Meteor Press Publishing House, 2008
10. Smeureanu, I., Dardala, M., Reveiu, A. "Visual C# .Net", CISON Publishing House, Bucharest, 2004
11. Soava, G. "Object oriented programming", Universitaria Publishing House, Craiova, 2008
12. Soava, G., Mehedintu, A. "Economic Informatics Systems", Universitaria Publishing House, Craiova, 2009