ASPECTS ON TEACHING / LEARNING WITH OBJECT ORIENTED PROGRAMMING FOR ENTRY LEVEL COURSES OF ENGINEERING

Clara Amelia de Oliveira

Professor – INE/CTC/UFSC. MSc.

clara@inf.ufsc.br

Marcos Fernando Conte

Student – Fellowship Holder of CNPq - Brasil

conte@inf.ufsc.br

Bernardo Gonçalves Riso

Professor - INE/CTC/UFSC. Dr.

riso@inf.ufsc.br

Universidade Federal de Santa Catarina – UFSC

Centro Tecnológico / Departamento de Informática e de Estatística – CTC / INE

Curso de Bacharelado em Ciências da Computação - CCO

C.P.: 476   CEP: 88.040 – 900   Florianópolis, SC.

Fone: (048) 331-9498   Fax: (048) 331-9770

ABSTRACT

This work presents a proposal for Teaching/Learning, on Object Oriented Programming, for Entry Level Courses of Engineering and Computer Science, on University .

The philosophy of Object Oriented Programming comes as a new pattern of solution for problems, where flexibility and reusability appears over the simple data structure and sequential process manipulation.

This approach leads with themas that includes all , data and methods/process in their internal levels named classes.

Object Oriented Programming , an approach nowadays more used for software engineering development, is a real challenge to be teached in early years of university disciplines, but, the waited results are: development of creativity , reach of philosoph principles of reusability, taking advantage of each part of work already done .

The proposed approach is essentially interdisciplinar, in aspects of linguistics of communication in texts,and,in technical aspects of programming submited to the high philosophal aspects of abstraction and hierarchy of mind.

An example in Object Pascal language is presented showing a proposed standard information for projects for teaching in entry level of university courses including thema discussion,purpose of programm, aspects of programming language and aspects of Object Oriented theorie.

1. INTRODUCTION

"When two persons walk on the same garden, each one sees their own garden. " - Oriental proverb.

The Object Oriented Programming (OOP) is a proposal, nowadays, more used for Software Engineering development, an specialized area in Computer Science. Their principles, based on abstraction, reusability, encapsulation and inheritance, bring to a natural one, that is integration of parts. Today, several problems are due to parts that have no integration with other ones, and, they don't work together, causing lots of garbage generation tasks. So, the proposal here is to offer, to the students of early years of university, an opportunity to think the usual problems with another point of view.

Then, a proposal for Teaching/ Learning (T / L) with Object Oriented Programming for beginners, will be discussed, through questions that are set, as follows.

2. WHAT MEANS T / L, THIS COUPLE TOGETHER, FOR THE AUTHORS OF THE PRESENT PROPOSAL ?

Teaching / Learning (T/L) is understood as a process that involves people, where each member have self rhythm, and, it is also an environment where everyone can learn or teach in accordance to each specific moment. For this, it is required flexibility and creativity because of the presence of diversity in aspects as personal styles of learning, as for example Kolb tests suggest [1] and personal possibility of changing self internal experience for learning. These two aspects are helped to reach, on pedagogical activities, employing also a proper linguistics of communication that leads with the people communication channels , as have Bandler and Grinder suggested, [7],[8] .

The challenge of teaching is to put together the theory of communication through linguistics, with the T / L environment, plus the technical information of the specific matter.

3. WHAT MEANS T / L OBJECT ORIENTED PROGRAMMING (OOP), FOR ENTRY LEVEL COURSES IN COMPUTER SCIENCE AND ENGINEERING ?

T / L, also in this context, is a special process, because, it leads with several aspects that goes from technical information till communication.

The theory of Object Oriented Programming (OOP) is based on representation of real world, employing the abstraction principle and basic abstract operations. So, at the end, to lead with OOP means to modeling, or, same way, to filtering what is important for that moment, but making this, in agree with general principles of hierarchy of mind, that support future expansions of the proposed theme of study.

The challenge here is to put together aspects as theory of OOP (art of modeling the proposed theme of study), data and logical structures of computational programs, offered from a chosen computational language, syntax commands of this language, and also the information from computational environment.

Aspect to be considered:		**How to manipulate the respective aspect:**
1. THEORY OF OOP	→	Abstract operations
2. DATA STRUCTURE	→	Abstract data type (Class), simple, array
3. LOGICAL STRUCTURES	→	Single sequence, repetition, conditional
4. COMMANDS AND SYNTAX	→	Computer language. Ex: Object Pascal
5. ENVIRONMENT	→	Environment Commands . Ex: Delphi

Figure 1: *Basic Considerations to Modeling/Programming in OOP Paradigm.*

4. WHAT IS MODELING ON OOP PARADIGM ?

Modeling is the science, and the art, right in this order, of representation. OOP search representation with harmony due to philosophic principles of abstract operations.

The programmer lead with representation of something that must be solved, developing schemes and algorithm. Representation is made with the modeling of part of the reality. This representation is an art, to be developed because the natural philosophic principles must be respected to make possible a development of a solution that follows the OOP paradigm proposal.

The challenge here is put, for T/ L in a interdisciplinary context, a kind of methodology that offers information of modeling, and conditions to learning, in a gradual process.

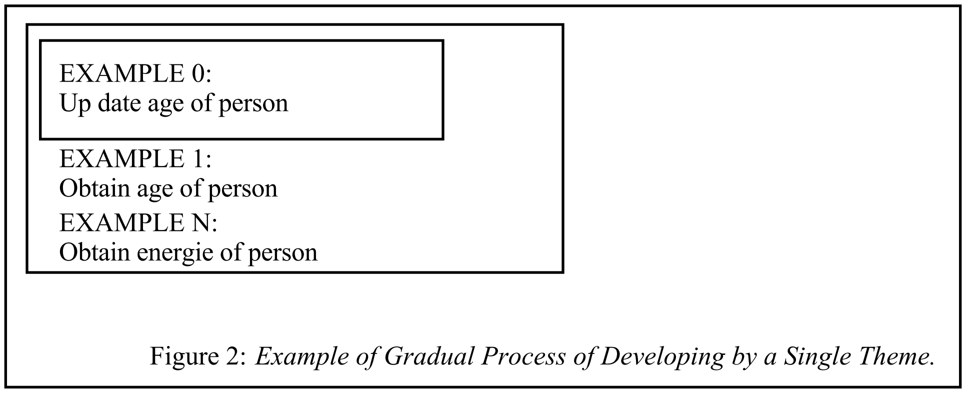


Figure 2: *Example of Gradual Process of Developing by a Single Theme.*

5. THE SCIENCE OR THE ART OF MODELING?

This is the question.

To reach the art state, should also Wolfgang Amadeus Mozart music study, with a teacher, in his early years of age, or, could he not write his brilliant ideas. So, each other learner also must begin with practical simple examples to be invited to perceive the philosophic principles of modeling in the OOP paradigm.

The OO literature, normally, approachs the art of modeling, with themes that contain a too complex grade of details operations for one that have, no experience about, in an airport, in an industry, or in an airplane factory, for example.

So, it's not easy to decide what to do, in a field you have no experience before, and, already more difficult to make the ´ how to do ´, when the ´ what to do ´ is not yet full understood.

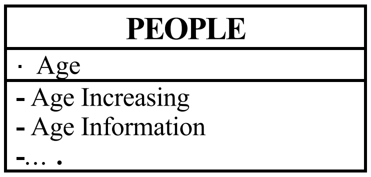
But this must to be made as a process, beginning fast like cooking receipts, and developing on the direction of the wide question in a continuous natural flux.

6. WHAT'S ABOUT THE PROPOSED METHODOLOGY ?

The proposed methodology begins with very very simple questions in a general theme. To make this, it is used the abstraction principle again. The spirit of this general principle is, the idea of ´ Simple As Possible ´, as had brilliancy Albert Paul Malvino already presented, as pedagogic method, in his book named ´Digital Computer Electronics: An Introduction to Microcomputers´, wrote over the years seventy. This principle permits, also in the present proposal, that the computer language, and OOP modeling begginer, learns syntax rules, of the chosen language, and, kinds of operation, acquiring, slowly and simultaneous, notions of modeling. Of course, this is very special recommended for a context of T/L with beginners. More advanced courses presuppose, that people already know a computer implementation language and computer environment, so, the discussion of theme, can follow the same methodology but in a more sophisticated context of representation and modeling.

But pay attention, the high abstraction principle adopted for beginners is not synonymous of fragmentation. The teacher must think well the way of development of this simple idea beginning with a few number of classes just to reach, in a harmonic and continuous way, a plateau of classes that lead, for example, until numerical methods or statistical methods on engineering. A development, step by step, of a single chosen theme, make this possible.

EXAMPLE 0:



EXAMPLE N:

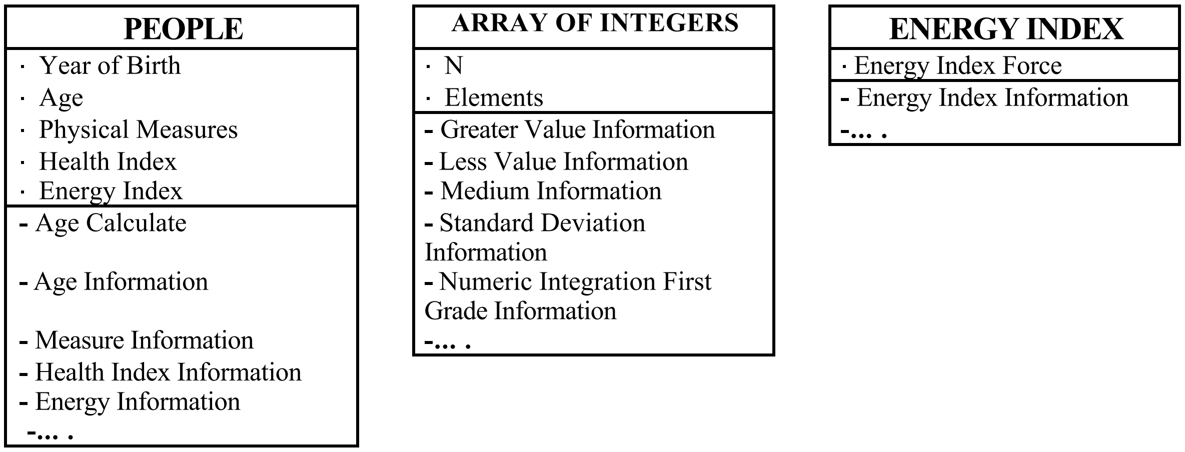


Figure 3: *Scheme of Some Classes to Indicate Different Stages of the Single Theme Development.*

7. WHAT, YET, TO SAY ABOUT A METHODOLOGY OF DEVELOPING A SINGLE THEME?

What is good, is that the several principles of OOP paradigm are reached in gradual and natural form, and the theme comes more and more wide until can man observe the work reusability. The programmer then notices, that he only develops and puts new details in an environment, called class, or develops new classes, several inherited from other still developed, profiting of this situation.

What, at the first view, is not good, but necessary, and at the end, positive, for T / L in entry level courses, when you work with a single theme during all the time, is, that you use simplification of a theme, to abstract complexity, just one that could cause difficults to beginners. The learner reachs each aspect in his right way or moment. The possible problem, with this proposal, is, that you normally must, after certain time, change some of the initially proposed data structures to more complexes ones and, consequent, change some classes, or part of them, because of the more sophisticated concepts, gradual presented. But, this is considered part of the methodology and, with this kind of work, you reach the great objective, that is, teaching for beginners, in computer, and simultaneous, in OOP paradigm.

-

8. WHAT'S ABOUT ABSTRACT OPERATIONS IN MODELING FOR T / L ENVIRONMENT?

The tree abstracts operations are: instantiation versus classification, generalization versus specialization and aggregation versus specification. The most important aspect here, is that each one can perceive that he naturally employs those operations to think, himself. Those operations are employed, as a natural way, to facilitate the mind flux searching a solution for the questions proposed.

Short, it can be said that all begin with the classification. There is a representation of the abstract data type, then following, natural appears, the general to specialized questions. Then, to describe each questions, they are decomposed in own parts, the decomposition. But, when they are too many parts, it is easily to aggregate them again, in new parts, to self organization, and so on. Those tree operations are the tree friends of the theme developer.

CLASSIFICATION	GENERALIZATION	AGGREGATION
X	X	X
INSTANTIATION	SPECIALIZATION	SPECIFICATION

Figure 4: *The Tree Abstract Operations for Representation and Modeling With OOP*

9. WHAT KIND OF THEME?

Pay attention, to choose a theme that is easy to develop in the direction of the discipline contents. For example, a theme for solve an equation is an specialized context. It leads with possibilities of methods implementation. It is viewed as a tool to solve a really general application proposal theme. The possibility of expansion must be always present in the proposal to be solved.

So, choose a real world wide theme,on the discipline context interest, discuss about questions to be solved, then, begin to work about.

10. WHAT'S THE WORK SEQUENCE IN MODELING ON THE PRESENT PROPOSAL ?

There are tree important steps to be considered: 'What to do'; ' How to do' and ' What to show'.

The work begins with 'What to do', the definition of tasks also named, the operations. This fundamental step of the whole work is made with discussion in the theme context and representation of scheme with proposed rules.

Follows the 'How to do', the implementation of tasks trough the named methods that are traduced by the code operations. This step is accomplished with algorithm construction.

Finally comes the 'What to show', the interface with the user, traduced by the input / output in visual or no-visual ways. This is a very important step but it live not alone, it depends of the quality of the first two steps.

WHAT TO DO?	Abstract operation
HOW TO DO?	Concrete method, implementation of code operation
WHAT TO SHOW?	Input / Output (Interface)

Figure 5: *Work Sequence in Modeling With OOP + Teaching / Learning Environment.*

11. STANDARD INFORMATION PROPOSAL FOR COMPUTER PROGRAMS MADE IN OOP PARADIGM FOR T / L CONTEXT

It will be presented below, part of a program in Object Pascal language, focusing of a kind of a set of standard information proposal, to be discussed, with members of T/L process, for entry level courses in computer language disciplines. The introductory program page, containing standard information, and, also tree classes are showed to illustrate the process. The selected classes are : PEOPLE (Pessoas), ENERGY INDEX (Ienergias) and ARRAY OF INTEGERS (Tveti).

A scheme, representing the tree selected classes, is presented in figure 3. A complete diagram, including relationships between the classes, can be developed, according to project methodologies as Booch, Rumbaugh, UML - Unified Modeling Language, etc, but, this kind of information is out of scope of the present paper. A complete diagram and program is, yet, available [14] .

The documentation, for this program example, first puts the theme title, and after, presents a description of purpose (what to solve in this theme for the moment).

After theme discussion, and its representation in a diagram, it comes a description, when necessary, of special information as aspects of domain theme, of modeling under object oriented concepts, of programming language, of data structure, of logical structures, and also of computer environment.

The present example focuses a numerical method application, and, it has adopted Object Pascal language in Delphi computational environment. It abstracts visual details aspects, using a crt application development for teaching objectives.

This process began with a very simple example, named, example zero. Example zero had only increased the age of a person. The presented example, or example named N, that calculate the numerical integral of an array of data, is a development on the theme that began with example zero. There is made ,as an option for teaching, a review of general aspects to bring , consciously, to the present level application example. Example N is suggested, for the last weeks of studies in OOP for an entry level course discipline in computer language, when special applications for engineering and computer courses, are welcome.

11.1 **Standard Information for T / L Strategy on Program Example**

11.1.1 THEME: Personal analysis (EXAMPLE N)

11.1.2 PROPOSAL: Calculate the person spent energy, based on force data X time.

Note. 1) Example N is a continuation from other previous examples that had began with example zero.

2) Here, it can be discussed, with students, about numerical integration operation and numerical integration implementation method, for example, integration of first degree.

11.1.3 Development of Classes Environment from Ex. 0 to N

a- DOMAIN CLASSES: PROGRAMAS (PROGRAMS), INTERFACES, PESSOAS (PERSONS), MEDIDAS (MEASURES), IENERGIAS (ENERGY INDEX), VETOR DE INTEIROS(ARRAY OF INTEGERS)

b- CLASSES TO BE ADDED: none

c- ATTRIBUTES AND METHODS TO BE ADDED:

ATRIBUTES: indiceDeEnergia (energyIndex)

METHODS: mudeDadosForca (changeForceData), informeIndiceDeEnergia (energyIndexInform)

11.1.4 Discussing and Reviewing Theoretical Aspects from Early to Present Example

a- ASPECTS OF OBJECT ORIENTED PROGRAMMING(OOP) TO BE DISCUSSED IN PRESENT EXAMPLE:

1) adding an object of energy index as attribute of person, so, person is free from the manipulation of details relative to this index.

2) presenting a vision of an application of a numerical method developed in a class that also can offer others mathematical and statistical ones.

b- ASPECTS OF OOP PRESENT IN THIS EXAMPLE AND ALREADY DISCUSSED IN PREVIOUS EXAMPLES OF THE CHAIN OF THE THEME: abstract data type, abstract operations of classification and aggregation X decomposition, encapsulation or information hiding, polymorphism, reusability.

11.1.5 Discussing and Reviewing Computer Language Aspects from Early to Present Example

a- ASPECTS OF OBJECT PASCAL LANGUAGE TO BE DICUSSED IN PRESENT EXAMPLE: none

b- ASPECTS OF OBJECT PASCAL LANGUAGE PRESENT IN THIS EXAMPLE AND ALREADY DISCUSSED IN PREVIOUS EXAMPLES:

types, reserved words, standard directives, expressions, modularity with procedure and functions.

11.1.6 Discussing and Reviewing Logical Structures Aspects from Early to Present Example

a- ASPECTS OF PROGRAMMING LOGICAL STRUCTURES: none

b- ASPECTS OF LOGICAL STRUCTURES ALREADY DISCUSSED:

simple, conditional, repetitive statements.

11.1.7 Discussing and Reviewing Delphi Environment Aspects from Early to Present Example

a- ASPECTS OF DELPHI ENVIROMENT: none

b- ASPECTS ALREADY DISCUSSED: Tobject class, visual and non-visual (crt application) possibilities.

11.1.8 Discussing About Next Proposition with the Students

NEXT STEP: reuse the developed class in a general program that is made for a group of persons, and, add a standard deviation method, for the persons energy index. This can be made also for an inherited class of person, that uses the person class.

11.1.9 Finally, Discussing with Students About Other Tasks Possibilities on the Present Example

TASKS TO STUDENTS:

Please, develop, for the index of energy attribute, a documentation considering the possibilities of future inherited classes that override some methods.

You can change the structure of array of integer class to array of real class, and, you can develop other methods.

After this, it can be discussed, with teacher, about tasks in other contexts.

12. EXAMPLE OF SOME IMPLEMENTED CLASSES OF PROGRAM ENERGY

Note, that for the below given examples of classes, only the new aspects for the full pedagogical sequence are inserted as commentary, and, for this presentation, they are also resumed.

12.1 Class Piece to Manipulate Persons Data:

```
unit Upessoa;
interface
uses umedida, uenergia, uvetor;
type
    PESSOAS  = class
    protected
      .............
      indiceDeEnergia:IENERGIAS;{**added attribute**}
    public
      .............
      function informeIndiceDeEnergia(quant:integer;intervalo:real):real;
    end;  {** added method operation **}
implementation
...
function PESSOAS.informeIndiceDeEnergia(quant:integer;intervalo:real):real;
begin  {** added method op-code **}
    informeIndiceDeEnergia:=indiceDeEnergia.informeIndiceDeEnergia(quant,
       intervalo); {** message to added object **}
end; end.
```

12.2 Class Piece to Manipulate Energy Index:

```
unit uenergia;
interface
uses utveti,uvetor;
type IENERGIAS = class
protected
```

```
   indiceDeEnergia:real;
   forcas:  TVETI;{**object/instance of array of integer class **}
public
   function informeIndicedeEnergia(quant:integer;intervalo:real):real;
end;
implementation
function IENERGIAS.informeIndiceDeEnergia(quant:integer;intervalo:real):real;
var energia:real;
begin
   energia:=forcas.calculeIntegralTrapezios(quant,intervalo);
   {** message to instance of the class of arrays of integers **}
   informeIndiceDeEnergia:=energia;
end; end.
```

12.3 **Class Piece to Manipulate an Array of Integer Data Structure :**

```
This kind of class is typical for a library to future reuse .
unit Utveti;
interface
uses uvetor;
type TVETI = class
...
end;
implementation
.....................
function TVETI.calculeIntegralTrapezios(nN:integer;h:real):real;
var s,i:integer;fx:veti;
    area:real;
begin  {** added method op- code **}
   for i:=1 to n do
     fx[i]:= elemento[i];s:=0;
   for i:= 2 to nN-1 do
     s:=s + fx[i];
   area:= h/2 *(fx[1]+ fx[nN]+ s*2);
   calculeIntegralTrapezios:=area;
end; end.
```

—

13. **CONCLUSIONS**

The Object Oriented Programming paradigm goes right on the direction of parts integration. Parts refer to solving of a whole proposal involving his static and dynamic aspects, represented by the attributes and methods, on the class implementation environment. This paradigm representation, favors, in a natural way, to reach interdisciplinary aspects of knowledge. This is real closed to the natural processes occurring on the real world.

In addition, the proposed pedagogic approach, centered on a high abstract level theme, favors flexibility and diversity, by expanding already done environments or developing new ones. This approach unified with OOP paradigm for programming, permits this fantastic flexibility due to OOP properties, as inheritance and late binding.

On the other hand, it is not casual that an environment of Teaching/ Learning must go just right in the same direction of parts integration. This is reached, using special linguistics of speech, and texts, characterized by diversity for motivation increment.

Finally , is it recommendable to create a physical environment, that implements ideas, making agreeable, the task of students of entry level courses. A few examples of some activities, are, painting or drawing charts for visual representation, or, yet, those that employ technological resources . There is a wide range of ideas, from simplicity to sophistication, that can be employed. And, this is a really nice work to be done to reach a future that each one wish for himself and for each other.

The present proposal of discussion, about Aspects on T/L with Object Oriented Programming for Entry Level Courses of Engineering, is a result of own searching and experimentation and students comments. Today, it is also supported by a Scientific Initialization Fellowship of CNPq- Conselho Nacional de Pesquisa.

Follows some suggestions of basic references per area of interest, for interested readers.

14. REFERENCES

This section contains a list of some basic bibliography on areas of interest on Teaching/ Learning environment and also in OOP paradigm and computer language programming and computational environment.

The areas of interest are subdivided in: Teaching / Learning, Neurolinguistics Programming, Object Oriented Programming, Object Pascal Language and Delphi Environment.

Teaching / Learning:

[1] FELDER, R., Matters of Style, ASEE Prism, USA,1996.

[2] GARDNER, H. Multiple Intelligences - The Theory in Practice, Howard Gardner Books, USA,1993.

[3] LINKSMAN, R. How to Learn Anything Quickly: an Accelered Program for Rapid Learning. Citadel Pr.USA,1996.

[4] ROSE, C. P. Accelerated Learning, DellBooks, 1989.

Neuro Linguistics Programming:

[5] ANDREAS, S., ANDREAS, C. Essência da Mente. Summus Ed.,S.P.,Brasil, 1995.

[6] BANDLER, R. Grinder, J. Patterns of the Hypnotic Techniques of Milton H. Erickson,M.D., Meta Publications, USA,1995.

[7] MOINE, D., HERD, J. Modernas Técnicas da Persuasão. Summus Ed.,S.P.,Brasil, 1988.

[8] OLIVEIRA, C. A., PACHECO, L., Apostila da Oficina Pedagógica em Tecnologia Educacional : Programação Neuro Linguística em Técnicas de Comunicação para Aulas e Apresentações, 7° Programa de Formação Pedagógica dos Docentes da UFSC, Florianópolis, Brasil, 1996.

[9] SZENÉSZI, G. Apostilas do Curso de Practitionner e Master em Programação Neuro Linguística.. MetaProcessos Avançados, Florianópolis, Brasil, 1996.

Object Oriented Programming:

[10] BOOCH, G. Object Oriented Design. B. Cummings. Prentice Hall, USA,1991.

[11] COAD, P., YOURDON, E. Object-oriented analysis, Yourdon Press,USA,1991.

[12] CONTE, M. F. Modelagem Orientada a Objetos para a Análise do Tema: Identidade de Pessoas Utilizando Objetos de Biblioteca. Monografia. UFSC, Florianópolis, Brasil, 1997.

[13] CONTE, M. F. Modelagem Orientada a Objetos para a Análise do Tema: Identidade de Pessoas Utilizando os Conceitos de Herança, Polimorfismo e Vinculação Tardia. Monografia. UFSC, Florianópolis, Brasil, 1997.

[14] CONTE, M. F. Modelagem Orientada a Objetos para a Análise do tema Identidade de Pessoas Utilizando Intgração Numérica. Monografia.UFSC, Florianópolis, Brasil, 1997.

[15] MARTIN, J., ODELL, J. Análise e Projeto Orientado a Objetos. Makron Books,S.P.,Brasil, 1996.

[16] MEYER, B. Object Oriented Software Construction. Prentice Hall,Great Britain, 1988.

[17] RUMBAUGH et alli. Object-oriented Modeling and Design, Prentice Hall, USA,1991.

Delphi environment and Object Pascal:

[18] CONTE, M. F. Documentação e Análise de Programas para o Ensino/Aprendizagem com Interface Visual.Monografia. UFSC, Florianópolis, Brasil,1997.

[19] RUBENKING, N. Programação em Delphi. Berkeley, S.P.,Brasil, 1995.

[20] WARNER, S., GOLDSMAN, P. Delphi 2.0 em exemplos. Makron Books,S.P.,Brasil, 1996.