

IFTE 2016 : 2nd International Forum on Teacher Education

Features of Professional Teachers Training of Informatics in a Programming Course

Tatjana U. Gainutdinova^a, Olga A. Shirokova^{b*}

* Corresponding author: Olga A. Shirokova, oshirokova@mail.ru

^a Kazan (Volga region) Federal University, Kremlyovskaya Street, 35, Kazan, Russia, oshirokova@mail.ru, tel. 8 (843) 2337160

^b Kazan (Volga region) Federal University, Kremlyovskaya Street, 35, Kazan, Russia, oshirokova@mail.ru, tel. 8 (843) 2337160

Abstract

<http://dx.doi.org/10.15405/epsbs.2016.07.6>

When selecting learning tasks, the development of object-oriented techniques in programming needs to be considered, demonstrating different approaches in solving the same problem. Students should be taught how to use knowledge in real situations, to expand the scope of possible applications of object-oriented programming. The approach to the sum propositioned in this article gives an opportunity to talk about real chance of the process' automation of difficult mathematical structures designing on the ground of the following stages: establishment, formalization and modeling of objects of structure, allocation of abstractions of the object's class and the methods; establishment of the communications between objects based on interaction and inheritance of classes; object-oriented design – distribution of classes on modules; creation of models for development of visual projects; - creation of effective algorithms of the sum making methods of the classes entering the project; implementation of the project in one of the chosen object-oriented programming languages; carrying out analysis of results of implementation of the object-oriented project.

© 2016 Published by Future Academy www.FutureAcademy.org.uk

Keywords: Object-oriented design, object-oriented programming, design, polymorphism, inheritance.

1. Introduction

Aims and goals of the professional improvement of future teachers of informatics on the basis of competence approach necessitate an in-depth study and application of innovative educational technologies.

The federal state educational standards of higher professional education involve a wide use of information and communications technologies in teaching students.



The course of programming is one of the major courses among disciplines of physics and mathematics. State educational standards of higher professional education in the major “Informatics” provide a studying program within the object-oriented structure. The approaches, which were still used, did not allow a fast developing information technology of visual programming in a proper degree. In order to improve the quality of teaching students, it is necessary to pay great attention to the study of various object-oriented programming environments in practice when developing visual projects.

However, the problem of low level skillfulness of students in initial stages of programming should be noted. Despite the fact that the study of the basics of structured programming is a part of the state educational standard of general and secondary education in computer science, a significant part of the freshmen has no clue about programming. However, such situation is fairly stable in recent years. Students, who have studied the basics of programming in the school course of informatics, are also experiencing difficulties in studying programming at a tertiary level.

That’s why it is necessary to immediately determine the set of the most important sections for studying programming in a continuing education, illustrating them with examples using high-level languages. The study of object-oriented programming (OOP) is overdue, this generates firstly from the operating skills with on-screen representations of information objects and models (Darakhvelidze, P. & Markov, E., 2005; Ivanov, G., 2011; Kvatrani, T., 2001). Secondly, OOP has a leading position in the development of software applications in their professional activities. The object-oriented approach is now leading in the development of information technology at the professional level.

The basic principles of OOP are inheritance, encapsulation and polymorphism. In accordance with the principle of inheritance, any descendant class can be generated from another class, this is indicated in its declaration of the parent class name. The offspring class automatically inherits the fields, methods, and properties of its parent and can add new ones. Inheritance mechanism allows you to create a hierarchy of classes.

Polymorphism - a property class that gives the opportunity to solve similar problems using different ways. As part of OOP, the behavioral properties of the class are defined by the set of its constituent practices. Descendants who are missing parent properties could be given certain properties, changing the algorithms and methods of inheritance. To change the method it is necessary to override it in the offspring, that is, declare a descendant of the same name in the process. As a result, object-parent and offspring-object, will be two derivatives of the method, and have different algorithmic basis, and therefore, contain different object properties. This is called polymorphism of objects.

1. Formulation of the problem

When teaching students the object-oriented style of programming, it is necessary to develop the ability to set the problem, identify objects and abstraction of the subject area, structure them by using the object-oriented techniques in programming.

Students are encouraged to develop visual projects during the learning of OOP techniques in the course - Programming. Among many other projects, it is proposed to create the project «University».

This article proposes a technique, which is based on the design of complex systems and structures; it is carried out using the following steps:

1. Establishment, formalization and modeling the structure of the object, allocation of abstractions of the object's class and determining the methods to be used.
2. Creation of models for the development of visual projects;
3. Establishment of communications between objects, based on interaction and inheritance of classes;
4. Object-oriented design – distribution of classes into modules.
5. Creation of effective algorithms and methods of entering the project class;
6. Implementation of the project in one of the chosen object-oriented programming languages;
7. Carrying out analysis of results of implementation of the object-oriented project.

2. Method of creating a project

Consider the methodology of the process of object-oriented designs and programming by creating a «University» project; during project development, it is necessary to use such basic principles as inheritance and polymorphism of the object-oriented programming (Shirokova, O., Training in programming on the basis of the community and distinction of the principles., 2015).

3.1. Determination, formalization and modeling of object structures, the allocation of abstractions: object classes and their methods

The development of the «University» project began, in accordance with the first stage of the proposed methodology.

When you create classes that are needed to solve problems, it is necessary to describe the fields and methods: students and teachers. The next step is to create two classes: T-stud and T-prof. Now we need to highlight the differences and similarities between the fields and methods of these classes. Common fields for the objects described by the classes are: name (f-name: string), address (f-adress: string), ID-ID (userID: string), phone (phoneNumber: integer), and the various fields are: the group number (f-gr: integer) and the name of the department (f-dep: string). Common methods are the constructor creation and the info function.

3.2. Creating models for the development of visual projects

This second stage involves the development of a model for visual project «University». The class diagram model (Figure 1).

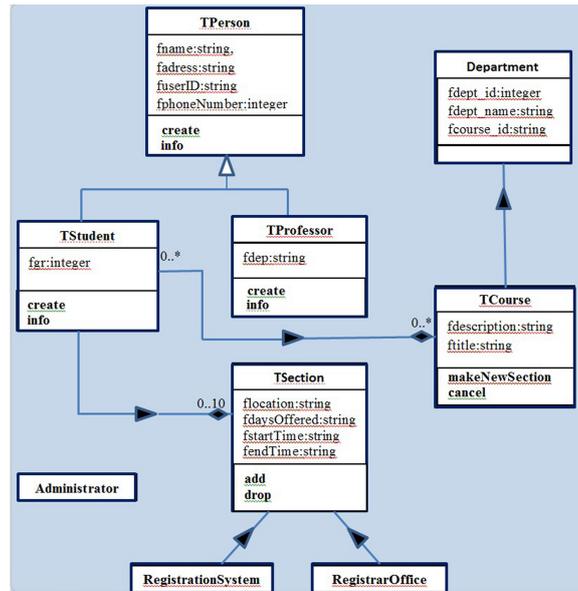


Fig.1. The class diagram model.

3.3. Determination of relationships between objects, based on the interaction and inheritance of classes

When developing the project in the third stage, you need to determine the relationship between objects, based on the interaction and inheritance of classes. You can use different approaches for solving this problem.

Due to the decomposition of the identified fields and methods that are common to classes, it is advisable to create a T-person class, which serves as the base for the two heirs: T-stud and T-prof classes.

The part of the project that includes a class diagram is the main module that generates and displays a list of students and teachers.

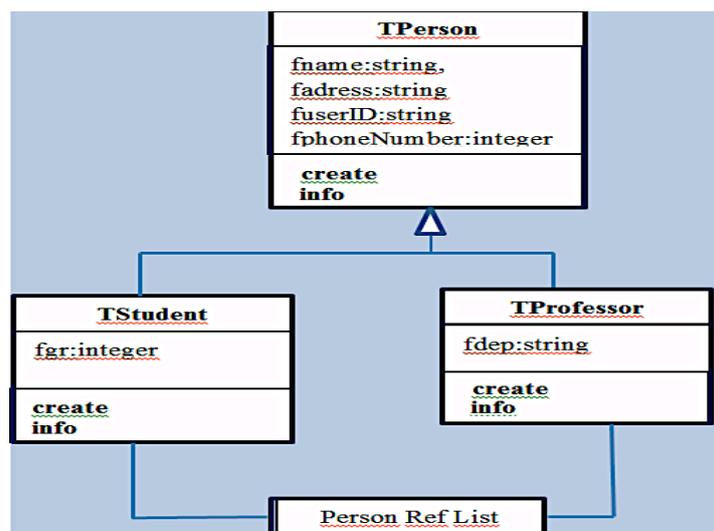


Fig.2. The main module diagram.

The constructor creator will carry out the function of recording the values in the field. Note, however, that the constructors in the classes of the student and the teacher should be different, except when setting the values to be recorded in the f-name, f-dep and f-gr. Creating a static constructor, is carried out by the descendants of the base class for the creation of objects:

```
constructor T-person.Create(name:string; address:string; user-ID:string; phone-number:integer);
constructor T-stud. Create(name:string; address:string; user ID:string; phone-number:integer;
gr:integer);
constructor T-prof. Create(name:string; address:string; user ID:string; phone-
number:integer;dep:string);
```

For us it is important to use info methods with a view on the object passed to it, if it is a descendant of the class T-Person, that is, info will be overlapped in the descendants of T-stud and T-prof. This allows descendant classes to produce a dynamic method replacement of its own.

3.4. Object-oriented design - the distribution of classes in modules

In the fourth stage of the development of object-oriented project, we will create a polymorphism module for implementation of the base and descendant classes using the Delphi system. Description module can also be made using C, C ++ systems and others. Imagine a fragment Polymorphism module:

```
Type
TPerson = class {base class}
fname: string;
constructor Create (name: string);
function info: string; virtual;
end;
T-stud = class (T-person) {descendant class}
fr: integer; { field for group number }
constructor Create (name: string; gr: integer);
function info: string; override;
end;
T-prof = class (T-person) {descendant class}
f-dep: string; ; {Field for the name of the department}
constructor Create (name: string; dep: string);
function info: string; override;
end;
```

Info methods will be dynamically overlapped using virtual directive in the descendants of the class-T-person: T-stud and T-prof; which will allow the descendant class to replace its own eponymous info

method in the module. Polymorphism concept provides a case of applying the object of use to the method that corresponds to the class object.

3.5. Creating efficient algorithms for solving the problems that make up the methods of classes included in the project, and the realization of one of the selected languages.

The fifth stage involves the creation of algorithms for solving problems that make up the class methods, included in the project.

This is done by defining the dynamic method info for each class individually:

```
function T-person.info:string;  
begin  
result:='';  
end;  
function Tstud.info:string;  
begin  
result:=fname+' group '+instructor(fgr);  
end;  
function Tprof.info:string;  
begin  
result:=f-name+' department '+f-dep;  
end;
```

Person Ref Sheet list can be written in array of T-person object class in the algorithm. The list takes the form: list: array [1..n] of T-person; {N} -size of list

It is possible to Declare the list as an array of T-person objects because the principle of polymorphism allows the assignment of a pointer to the parent class, and a pointer to a class-descendant. Therefore, the elements of the list array can be both objects of T-stud class, and objects of T-prof class.

This is the displays od a list by applying the info method to the list array elements:

```
St:= '';  
for i:=1 to szl do  
if list[i]<>nil then  
St:=St+list[i].info+#13;  
ShowMessage('Spisok:'+St); {Displays a list in the box messages }
```

During the execution of the program, each element of the array can contain an object of T-stud type, and an object of T-prof type.

3.6. Implementation of visual design at one of the selected object-oriented programming languages

In the sixth stage, design on one of the selected object-oriented programming languages is implemented. A visual application is also developed. (Shirokova, O., 2015). Window shapes will look like Figure 3.

The project includes visual components such as: Label1, Label2, Edit1, Edit2, Button1, Button2. GroupBox1 - a component that brings together a group of components with similar meaning. In this case, it has two dependent switch : RadioButton1, RadioButton2.

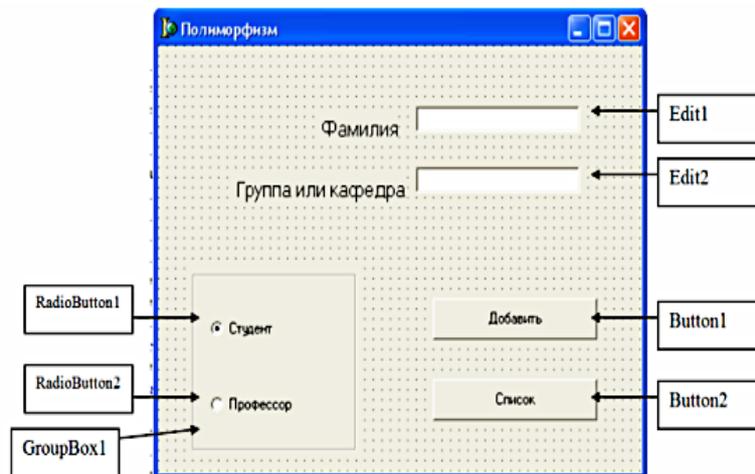


Fig.3. Window shapes of visual application.

3.7. Analysis of the results of the implementation of object-oriented designs.

At the seventh stage of the analysis, the results of the implementation of «University» object-oriented design would be analyzed.

The results of the project are shown in Figure 4:

T-Form1.Button1Click procedure is performed using the button "Add" to create the list [n]of objects in T-stud class or T-prof class. The class of objects to be created is determined by the state of the Radio Button switch . Setting the switch to the Student defines the T-stud class,and setting it to the position of the Professor defines the T-prof class .

T-Form1.Button2Click procedure is performed using the "List ", it applies the info method to each element of the list [n] array as an object to form a displaying list of students and teachers.

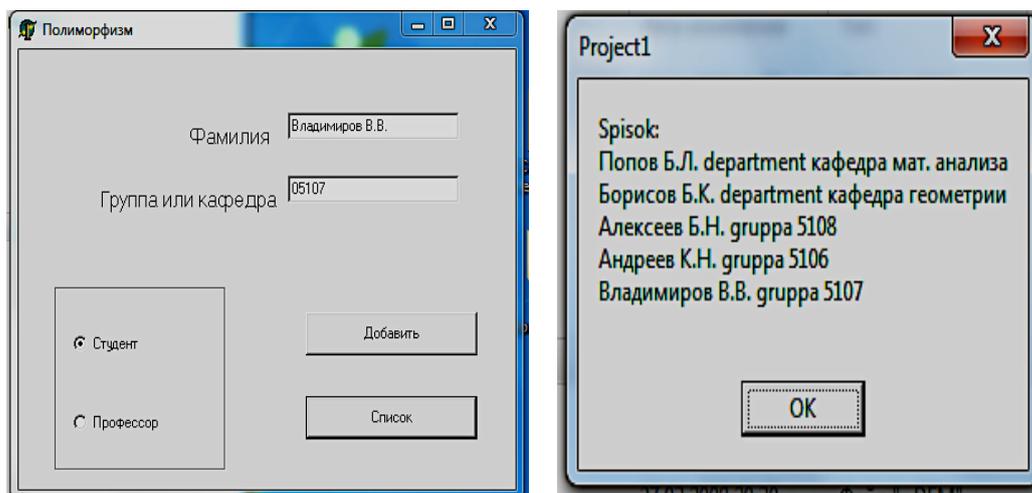


Fig.4. The results of the project.

3. Analysis of the study results

This study focuses on the problem of polymorphism in the implementation of «University» system project. Note that when using object classes, T-stud and T-prof used the principles of inheritance and polymorphism. These principles are the basis of object-oriented university system and it implemented the project by creating a base class T-person. Description of the interaction of the base class and its descendants is a key issue when creating system «University».

Therefore, the description of the main unit is fundamental for the creation of this great system.

The ability of students to think "objectively" is formed in the development of visual projects of standard components of object-oriented programming systems.

This methodology is a phased project involved in the creation of an increasing scope and roles of independent works of students and the widespread use of active and interactive learning methods.

4. Conclusion

This article highlights and discusses the features of the study course - technology of object-oriented programming, as one of the main components of the formation and development of professional competencies, as well as the ability to act on the basis of the available skills, knowledge and experience in a particular area of professional activity.

Creating an object-oriented system of modeling projects and structures, contributes to the formation of students' skills in formalization, abstraction, separation, structuring and implementing of objects. Ability to apply software development models to create software products, as well as the use of software systems modeling tools are formed among students.

It is necessary to pay great attention to the creation of object-oriented projects in different environments to improve the quality of teaching "Programming" course.

In accordance with this methodology the student must acquire the ability to independently develop a project while learning OOP. Practical experience shows that efficiency of studying programming increases: theoretical knowledge is acquired at its best and the skills of formalization, design, modeling and programming are also well practiced.

References

- Darakhvelidze, P. & Markov, E. (2005). Programming in Delphi, 7. 784.
- Ivanova, G. (2011). Programming Technology: a textbook, 336.
- Kvatrani, T. (2001). Rational Rose 2000 and UML, 176.
- Shirokova, O. (2015). Training in programming on the basis of the community and distinction of the principles. *Modern problems of science and education*, 1, 1757-1762.
- Shirokova, O. (2015). An object-oriented projects for solving mathematical problems. *XI International scientific-practical. Conf. Object Systems - 2015*, 15-23.