

Ruslan I. Bazhenov,
PhD, associate professor;

Fedor G. Dzikovsky,
student;

Olesya Ya. Dubei,
student;
Sholom-Aleichem Priamursky State University

The Teaching of Object-Oriented Approach on Programming Language F#

Keywords: *teaching, object-oriented programming, F#*

Annotation: *The paper describes the teaching methodology of object-oriented approach (OOA). The algorithm of methodology study is presented and illustrated: 1) the repetition of OOA concepts; 2) examples of OOA declarations; 3) the implementation in the console; 4) the implementation in Visual Studio windowed mode; 5) the implementation of individual tasks.*

Язык программирования F# поддерживает различные функциональную (основная парадигма), императивную, объектно-ориентированную парадигмы. F# позволяет писать как пользовательские приложения, так и библиотеки, которые возможно использовать в других приложениях, создаваемых на технологии .NET. Обучению программированию представленным парадигмам в алгоритмических языках посвящено множество исследований (1, 3, 8). Рассматриваемый язык является молодым и существует достаточно много методическим проблем в его преподавании.

Хотя функциональная парадигма является основной в языке F#, необходимо уделять внимание объектно-ориентированному подходу (ООП), так как эта методология используется большим количеством программистов. В образовательной программе «Информатика и вычислительная техника» Приамурского государственного университета имени Шолом-Алейхема ведется курс «Логическое и функциональное программирование» на котором студенты изучают F#. Поэтому была сформирована задача по проектированию лабораторной работы, в которой можно было наглядно показать принципы объектно-ориентированного подхода в языке F#.

Содержательную сторону проблемы можно почерпнуть из пособий российских авторов Д.В.Сошникова (6), Ан.В.Шамшева, В.В.Ворониной (4). Описание технологии использования ООП в F# приводят зарубежные ученые Т.Петричек, С.Смит и др. (2, 5, 7).

Для обучения представленной методологии авторы предлагают использовать следующий алгоритм: 1) повторение понятий ООП; 2) примеры объявления ООП; 3) реализация в консоле; 4) реализация оконного режима в Visual Studio; 5) выполнение индивидуальных заданий.

Приведем иллюстрации алгоритма.

1. Повторение понятий объектно-ориентированного подхода.

На данном этапе достаточно повторить со студентами понятия: класс, объект, полиморфизм, интерфейс, метод, инкапсуляция, наследование.

2. Примеры объявления ООП.

Показываем студентам, что первым шагом в построении объектной модели являются записи (рис.1).

```
type Point = { x : float; y : float }
let p1 = { x=10.0; y=10.0 }
let p2 = { new Point with x=10.0 and y=0.0 }
```

Рис. 1. Пример записи в F#

Далее переходим к описанию класса `_Massiv_`, который будет содержать в себе три метода его обработки: заполнение случайными элементами от 0 до 200; сортировка этих элементов методом «пузырька»; подсчёт среднего арифметического элементов массива. Описываем набор методов в виде записи (рис. 2).

```
type _Massiv_={_Fill_:unit->int[]; _Sort_Bub_: unit->int[]; _Average_:unit->int}
let Massiv2 n =
    let cou = n
    { _Fill_ = fun () -> for i = 0 to n - 1 do
                            x.[i] <- el.Next(200)
                        }
    ...
}
```

Рис. 2. Фрагмент описания класса через запись

Представляем другой способ описания класса (рис 3.).

```
type Massiv = None
with
    member M.Fill n =
        for i = 0 to n - 1 do
            x.[i] <- el.Next(200)
        x
    ...
end
```

Рис. 3. Фрагмент описания класса с методами

Дополняем объяснение еще одним способом реализации класса через описание его интерфейса (рис.4).

```
type Massiv_ = interface
    abstract Fill_: n: int -> int[]
    abstract Sort_Bub_: n: int -> int[]
    abstract Average_: n: int -> int
end
let Massiv1 =
    { new Massiv_ with
        member m.Fill_(n) =
            for i = 0 to n - 1 do
```


Для иллюстрации использования объектно-ориентированного подхода с библиотекой WinForms демонстрируем ее код (рис. 6) и вызываем разработанную программу (рис. 7).

```
// Кнопка для поиска среднего арифметического элементов массива
let But9 = new Button(Text = "Найти среднее арифметическое элементов
массива")
But9.Location <- new Point(5, 230)
But9.Width <- 300
But9.Height <- 25
But9.Click.AddHandler(fun __ ->
    Lbl11.Text <- ""
    let result8 = M3 |> List.map (fun _Massiv_ -> _Massiv_.Average_())
    Lbl11.Text <- Lbl11.Text + " " + string result8)
```

Рис. 6. Фрагмент кода программы с WinForms

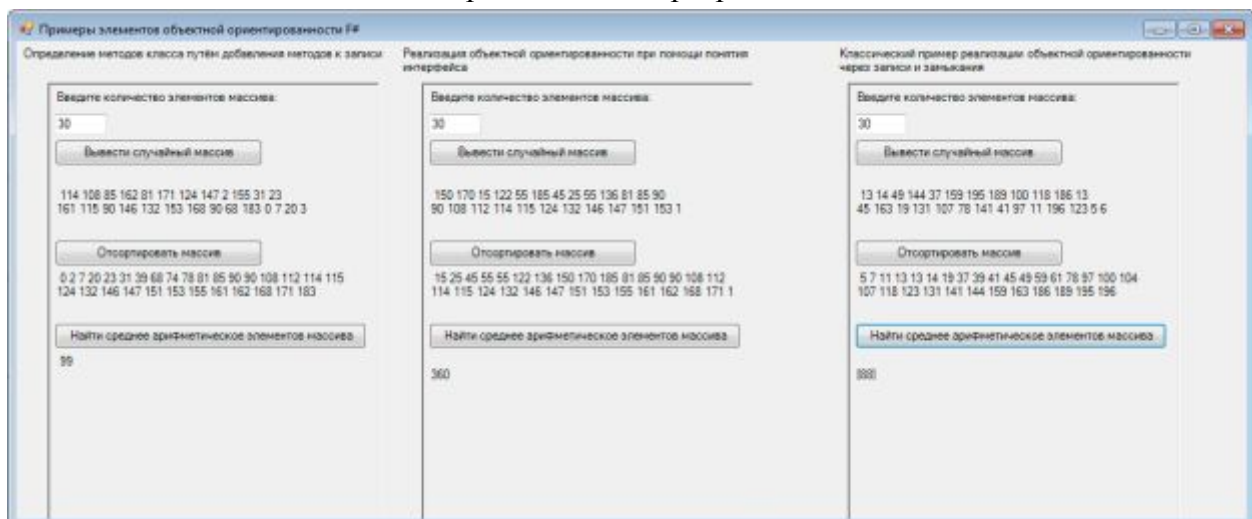


Рис.7. Выполнение примера в оконном режиме

5. Выполнение индивидуальных заданий.

Для закрепления знаний и получения умений самостоятельного программирования студентам предлагается решить задачи. Все задания необходимо выполнять с помощью библиотеки WinForms, или WPF (Windows Presentation Foundation). Приведем несколько примеров.

1) Написать класс при помощи добавления методов к записи, реализующий следующие методы: нахождение площади треугольника, заданного тремя точками; определение, является ли треугольник, заданный тремя точками равнобедренным; является ли этот треугольник прямоугольным.

2) Написать класс при помощи понятия интерфейса, реализующий методы нахождения объёмов различных тригонометрических фигур: конуса, заданного радиусом основания, координатой центра основания, координатой вершины; пирамиды заданной четырьмя точками; цилиндра заданного радиусом основания и координатами центров оснований.

Предлагаемая методика обучения объектно-ориентированному подходу на языке F# показала свою эффективность. Студенты свободно адаптировались к использованию описываемой методологии в новом для себя языке программирования. Данный способ

позволил показать возможность совместного использования технологий разработки программного обеспечения в среде Visual Studio, на C# и F#. При написании выпускных квалификационных работ в 2011-2013 гг. примерно треть студентов избирали для реализации объектно-ориентированную связку C# и F#.

References:

1. Hedin G, Bendix L, Magnusson B. Teaching extreme programming to large groups of students. *Journal of Systems and Software* 2005;74(2):133-146. [\[Google Scholar\]](#)
2. Petricek T, Skeet J. . *Functional programming for the Real World with examples for F# and C#*: Manning: Ger; 2007. [\[Google Scholar\]](#)
3. Jm RC, Balcells AC, Estévez AM, Moreno GJ, Mj FR. A game-based approach to the teaching of object-oriented programming languages. *Computers & Education* 2014;1:83-92. [\[Google Scholar\]](#)
4. Shamshev AB, Voronina VV. *Functional programming language F #: Tutorial*. Ulyanovsk: UISTU; 2012. [\[Google Scholar\]](#)
5. Smith C. *Programming F#*. O'Reilly: unknown; 2010. [\[Google Scholar\]](#)
6. Soshninov DV. *Programming F#*. M. DMK Press; 2011. [\[Google Scholar\]](#)
7. Syme D, Granicz A, Cisternio A. *Expert F#*. A-Press: unknown; 2007. [\[Google Scholar\]](#)
8. Wang Y, Li H, Feng Y, Jiang Y, Liu Y. Assessment of programming language learning based on peer code review model. *Implementation and experience report: Computers & Education* 2012;59(2):412-422. [\[Google Scholar\]](#)