

МЕТОДИЧЕСКИЕ ПОДХОДЫ К ОБУЧЕНИЮ ОБЪЕКТНО-ОРИЕНТИРОВАННОМУ ПРОГРАММИРОВАНИЮ В СИСТЕМЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ ДЕТЕЙ (НА ПРИМЕРЕ КУРСА ДЛЯ 8–10 КЛАССОВ)

В статье рассмотрено процесс обучения объектно-ориентированному программированию без предварительных знаний алгоритмических конструкций структурного программирования учащихся 8–10 классов в системе дополнительного образования детей.

Ключевые слова: объектно-ориентированное программирование, дополнительное образование детей.

В условиях социально-экономической трансформации и информатизации современного общества одним из приоритетных направлений является процесс внедрения и использования средств информационных и коммуникационных технологий (ИКТ) во все сферы жизнедеятельности человека. Одной из фундаментальных отраслей научного знания в данном процессе, способствующей формированию системно-информационного подхода к анализу окружающего мира, изучению информационных процессов, методов и средств получения, преобразования, передачи, хранения и использования информации, является в общеобразовательной средней школе учебный курс «Информатика и ИКТ».

Учитывая современные тенденции развития информатики, ведущим и наиболее перспективным направлением, по мнению Андросова Е.Г., Ивановой Д.С., Петровой Ю.А., Петрова А.Н., Петухова А.Ю., Сидорова В.А., Хачева Г.Ю. и др., является объектно-ориентированное программирование [1, 5, 6, 7, 8].

В современной практике преподавания примерная программа среднего (полного) общего образования по информатике и информационным и коммуникационным технологиям на раздел практика программирования предусматривает 16 часов. В школьных учебных программах объектно-ориентированные языки либо вообще не рассматриваются в учебно-методических комплексах (Шафрин Ю.А., Макарова Н.В.), либо программа предусматривает основы программирования в достаточно сжатый период времени и объем часов (Н.Д. Угринович). В этой связи, у выпускников школ, определяющих свою будущую профессию со специальностями, связанными с изучением информатики, возникают труд-

ности при изучении объектно-ориентированного программирования, в силу недостаточной базовой подготовки в области программирования. Оптимальным решением данной проблемы считается включение объектно-ориентированных языков программирования в учебную деятельность учреждений системы дополнительного образования детей.

Как показывают исследования Герасименко С.А., Ермилиной Е.В., Занозинной Г.В., Щетинского Ю.А. и др., дополнительное образование детей по информатике дает возможность расширить знания по предмету, усилить мотивацию учащихся к творческому процессу и развить интерес к самостоятельному поиску решения практических задач [2, 3, 4, 9]. Отличительной особенностью формирования групп системы дополнительного образования детей является обеспечение эффективного процесса обучения школьников с разным уровнем знаковой подготовки, мотивационной и возрастной специфики. Поскольку одним из основополагающих принципов системы дополнительного образования детей считается обеспечение образовательных потребностей всех школьников независимо от возраста по добровольно выбраным ими направлениям, центральное место отведено организации диагностических процедур, по результатам которых формируются учебные группы, состоящие из микрогрупп школьников по их способностям.

При этом основными социальными функциями дополнительного образования детей являются обеспечение занятости детей и их интеллектуальное развитие в свободное от основных занятий в школе время, профессиональное самоопределение и профилактика детской преступности.

Анализ учебных программ по информатике в учреждениях общего среднего образования и системы дополнительного образования детей подтверждает факт их тесного переплетения. Программы по информатике в учреждениях дополнительного образования детей основаны, как правило, на содержании, которое либо перестало быть обязательным в школьном курсе, но остается важным на этапе социального и профессионального развития личности учащегося, либо включено в школьный стандарт в объеме, недостаточном для решения проблемы профессиональной ориентации школьников (в частности, при обучении объектно-ориентированному программированию).

В настоящее время востребованными остаются дополнительные занятия по обучению объектно-ориентированного программирования среди школьников. Значительное внимание среди специалистов в области информатики (М.А. Зайцева, Ю.В. Калмыков, Е.В. Келлер, Н.В. Коробков, Т.Б. Корнеева, Е.К. Кравцова, Л.В. Краева, А.П. Пермякова, В.А. Сидоров и др.) уделяется обучению объектно-ориентированным языкам после получения базовых знаний структурного программирования. Такой подход можно назвать классическим и он требует значительных временных затрат. Кроме того, возникает барьер на этапе перехода от сравнительно простого интерфейса оболочки структурного языка программирования к сложному виду среды объектно-ориентированного языка.

Решением указанных проблем является применение в практике преподавания объектно-ориентированного программирования научно-методического подхода, основанного на изучении курса без предварительного изучения структурной парадигмы. Учебный курс может быть реализован на основе алгоритмических конструкций языка, на примерах работы с объектами. В этом случае даже в условиях сокращения времени обучения основам программирования у школьников не возникают барьеры при переходе от одной программной среды к другой.

Структура разработанного нами учебного курса основана на трех ключевых модулях: 1) лекционные материалы; 2) лабораторные практикумы; 3) методические рекомендации. Каждой теме теоретического раздела должна соответствовать лабораторная работа, предназначенная для закрепления полученных теоретических сведений на практике. Практические задания должны содержать базовые задания и

предложение с методическими рекомендациями по разработке индивидуального мини-проекта на интересующую ученика тему с целью закрепления полученных знаний, являющихся обязательным минимумом для выполнения каждым школьником. Кроме того, обязательен раздел с заданиями повышенной сложности.

Разработанный нами курс, направленный на изучение объектно-ориентированного программирования в системе дополнительного образования детей 8-10 классов, основан на блочно-модульной структуре, включающая базовый и профильный блоки обучения. Базовый блок обучения ориентирован на рассмотрение основ объектно-ориентированного программирования; профильный блок способствует расширению знаний в области основ объектно-ориентированного программирования. Базовый блок включает разделы: алгоритмы и исполнители, основные понятия объектно-ориентированного программирования, основные операторы языка, компоненты объектно-ориентированного языка программирования Delphi, создание собственных компонентов. Профильный уровень состоит из следующих тем: технология Drag&Drop, создание подпрограмм: процедур и функций, решение арифметических задач повышенной сложности, понятие технической записки, мультимедиа и графика, внедрение и связывание объектов, динамический обмен данными, обработка исключительных ситуаций. Определены формы, методы и средства обучения.

Остановимся на особенностях формирования учебных групп. Проведение диагностических процедур целесообразно строить на методах проведения компьютерного тестирования, анкетирования, интервьюирования, собеседования, выполнения практических заданий с целью выявления базовых умений и навыков и готовности к решению задач творческого уровня.

В процессе реализации учебного курса деление на учебные микрогруппы, как правило, происходит по результатам усвоения вводного модуля (через полтора-два месяца обучения). Школьники определенной микрогруппы получают: задания одинакового уровня сложности; индивидуальные задания; проекты по проблемам в зависимости от личных интересов школьника. Например, для ученика, увлекающегося историей родного края, может быть предложен проект, проверяющий знания пользователя в этой области; школьник, увлекающийся математикой, работает над справочником матема-

тических формул и т.д. Интересным фактом является то, что не всегда в микрогруппы попадают учащиеся одного возраста.

Контроль знаний должен основываться на методике диагностических процедур знаний, умений и навыков, построенных на критериях оценивания мини-проектов по окончанию каждой новой темы, а также итоговых проектов за полугодие и год. Благодаря использованию проектной методике, моделируется постоянная «ситуация успеха» для каждого участника учебного процесса, что способствует их высокой мотивации в обучении. Предусмотрена проверка теоретических знаний учащихся с помощью тестовых заданий и устных опросов школьников педагогом.

Для промежуточного контроля школьники готовят зачетные проекты. Во время защиты своих проектов они должны четко обосновать цель работы, дать анализ успешности или проблемных ситуаций по их достижению, предоставить экспертной группе программы по реализации данного проекта на будущее. В экспертную группу, как правило, входят школьники, изучающие программирование второй год.

На этапе завершения первого года обучения школьники создают индивидуальные итоговые проекты. Чаще всего итоговый проект является продолжением зачетной работы. Эти программные продукты проходят публичную защиту. Их оценивает независимое компетентное жюри, в состав которого могут быть включены педагоги учреждений дополнительного образования детей, студенты и преподаватели физико-математических факультетов города.

Второй год обучения предполагает самостоятельное рассмотрение представленных разделов школьниками, используя лабораторные практикумы и дополнительные информационные источники. К публичной защите, школьники создают программные продукты, которые оценивает экспертная группа педагогов учреждения дополнительного образования, председателем жюри приглашается независимый научный эксперт.

С целью формирования методологических и базовых умений воспитанниками учреждений дополнительного образования, в экспериментальной группе были использованы некоторые методические приемы, которые позволили получить ожидаемые результаты.

На одном из первых занятий учащимся предлагается рассмотреть основные понятия

объектно-ориентированного программирования. Для осмысления многогранности понятий и лучшего восприятия, понимания и усвоения учащимися системы дополнительного образования проводится семинарское занятие, на котором воспитанники выступают с сообщениями о различных определениях таких понятий как «объект», «ориентация», «программирование», «свойство», «метод», «событие», «инкапсуляция», «наследование» и «полиморфизм». Учащиеся находят определения предложенных понятий в энциклопедических, философских и других источниках и, обязательно, с точки зрения теоретической информатики. По окончании выступлений совместно с учащимися педагог делает выводы о том, как определить эти понятия относительно объектно-ориентированного программирования. Особо стоит подчеркнуть то, что перед домашним заданием по поиску информации с учениками проводится мастер-класс, на котором предлагаются основные положения по поиску информации в печатных изданиях и в интернет-источниках. В основном успех семинарского занятия достигается соревновательным моментом – воспитанник, нашедший наибольшее количество определений получает почетное звание «энциклопедиста».

Понятия «метод», «свойство» и «событие» в основном не вызывают не понимания. Педагог акцентирует внимание на том, что произошло какое-то событие, например, упала ручка. Методом устранения этого события служит действие по ее поднятию. Учащиеся быстро приходят к аналогии произошедшего с написанием программы, поскольку каждое событие должно быть обработано. Для обработки необходимо написать метод, то есть последовательность действий (алгоритм) для ответа на произошедшее событие.

Как правило, первые проблемы возникают в процессе изучения «условного оператора». Считаем эффективным параллельное изучение понятия «условный оператор», компонентов классов `TCheckBox` (флажки) и `TRadioButton` (переключатели). Важным моментом методического приема является акцентирование внимания школьников на использование флажков в среде программирования на этапах выбора собственного решения (да/нет или да/нет/не уверен).

Учащиеся должны убедиться на практике в корректности решения задачи и сделать проверку: при установке флажка текст в Метке 1 должен становиться жирным.

```

Procedure TForm1.CheckBox1Click
(Sender: TObject);
begin
  if checkbox1.Checked=true
  then Label1.Font.Style:=font.
style+[fsbold]
  else Label1.Font.Style:=font.
style-[fsbold]
end;
    
```

Таким образом, учащиеся делают вывод, что условный оператор позволяет проверить некоторое условие и в зависимости от результатов проверки возможно выполнение определенного действия.

На основе переключателей (TRadioButton) можно рассмотреть так называемое неполное ветвление, когда активна только одна из ветвей оператора. Для этого может быть предложена, например, такая задача: при установке переключателей текст в Метке1 должен становиться желтым, либо красным.

```

procedure TForm1.RadioButton1Click
(Sender: TObject);
begin
  if RadioButton1.Checked=true
  then Label1.Font.Color:=clyellow
end;
procedure TForm1.RadioButton2Click
(Sender: TObject);
begin
  if RadioButton2.Checked=true
  then Label1.Font.Color:=clred
end;
    
```

Структура неполного ветвления в общем виде:

IF <УСЛОВИЕ> THEN <ОПЕРАТОР>;

Для облегчения работы с оператором на начальном этапе работы с ним, можно предложить таблицу, что является ценным для учащихся:

<Условие>	<Оператор>
a>0	K:=0; k:=k+1; //количество
a<0	S:=0; s:=s+a; //сумма
a=0	P:=1; p:=p*a; //произведение
...	...

Методика работы с таблицей довольно проста: в структуру условного оператора IF <УСЛОВИЕ> THEN <ОПЕРАТОР №1> ELSE <ОП.2>; вместо <УСЛОВИЕ> и <ОПЕРА-

ТОР> используем нужные поля таблицы. В предлагаемые столбцы, можно вводить какие угодно условия и операторы, необходимые для решения задачи.

При изучении языков программирования у учащихся «безинициативной» группы возникают трудности с реализацией *алгоритмов решения задач*. Пусть алгоритм решения имеет 3 основных этапа:

1) ввод данных

Например:

```

a := StrToInt(Memo1.text); // занести в переменную a содержимое многострочного редактора Memo1, переведенное из строкового в числовую форму для выполнения математических действий.
    
```

```

b := StrToInt(Edit1.text); // занести в переменную b содержимое однострочного редактора Edit1, переведенное из строкового в числовую форму для выполнения математических действий.
    
```

2) проведение определенных действий, требуемых в условии задачи.

3) вывод результата или сообщения о проведенном анализе данных.

Вывод результата: при решении примера c := a+b; ответ c.

```

Memo1.text := IntToStr(c); // вывод результата решения примера в многострочный редактор Memo1 путем перевода ответа из числовой в строковую форму.
    
```

Вывод сообщения:

```

Edit1.text := 'Задача решена успешно!'; // вывод сообщения о проведенном анализе данных в многострочный редактор Edit1.
Обратим внимание, что в данном случае перевод сообщения из числовой в строковую форму не осуществляется.
    
```

Рассмотрим пример решения арифметической задачи: найти сумму положительных и количество остальных чисел из введенных x, y, z.

1 этап:

```

x := StrToInt(Edit1.Text);
y := StrToInt(Edit2.Text);
z := StrToInt(Edit2.Text);
kol := 0; sum := 0;
    
```

2 этап:

```

If x>0 then sum := sum+x else kol := kol+1;
If y>0 then sum := sum+y else kol := kol+1;
If z>0 then sum := sum+z else kol := kol+1;
    
```

```

3 этап:
Mem1.Lines.Add(IntToStr(sum));
Mem1.Lines.Add(IntToStr(kol));
Теперь подставляем каждый заготовлен-
ный этап на свое логическое место в программе.
Program <имя>;
Var x, y, z, kol, sum : integer;
{описание данных}
Begin
{1 этап}
  x := StrToInt(Edit1.Text);
  y := StrToInt(Edit2.Text);
  z := StrToInt(Edit2.Text);
  kol := 0; sum := 0;
{2 этап}
  If x>0 then sum := sum+x else
kol := kol+1;
  If y>0 then sum := sum+y else
kol := kol+1;
  If z>0 then sum := sum+z else
kol := kol+1;
{3 этап}
  Mem1.Lines.Add(IntToStr(sum));
  Mem1.Lines.Add(IntToStr(kol));
End.

```

Как показывает практика, усвоение понятия *цикла* вызывает трудности у большинства учеников. Считаем эффективным объяснение этого понятия совместно с изучением строкового типа данных, поскольку одним из вариантов использования цикла является работа со строками.

Методической находкой является опора на то, что отличительной чертой типа данных `string` является следующее: мы можем обратиться как к строке целиком, так и посимвольно, то есть к каждой букве в отдельности. Для этого как раз нам необходимо использовать оператор цикла.

При работе со строками чаще рекомендуем использовать цикл `For` (с параметром).

При работе со строкой запишем оператор цикла следующим образом:

```
for i:=1 to length(s) do (перебор по
символам от первого до последнего в строке s).
i – индекс, то есть порядковый номер каждой
буквы в строке.
```

Ввод и вывод строки осуществляется через стандартные компоненты ввода/вывода, такие как `Edit`, `Memo`, `LabeledEdit` и др. Но при проверке какого-либо условия, когда необходимо рассмотреть каждую букву в отдельности необходимо использовать оператор цикла.

Рассмотрим пример алгоритма решения задачи, в которой нужно подсчитать количество

вхождений символа «а» в строку. Используем компоненты классов `TEdit` и `TMemo` для ввода строки и вывода результата.

```

Var s: string; i, k: integer;
Опишем методы решения задачи:
Procedure TForm1.Button1Click
(Sender: TObject);
Begin
  s:=edit1.text; //ввод строки
  For i:=1 to length(s) do //идем
по строке
    If s[i]='a' then k:=k+1; //если
i-ый символ а, тогда увеличивай счетчик
k. s[i] – обращение к букве строки s.
    Mem1.text:=IntToStr(k) //вывод
результата
End;
End;

```

Особое внимание уделяем рассмотрению *массивов*, поскольку в школьном курсе математики не вводится понятие матрицы, учащиеся «безинициативной» и «социально-ориентированной» групп с трудом осваивают данную структуру. Для наглядности работы с массивами помогают компоненты классов `TStringGrid`, что позволяет визуально представить себе этот тип данных. А доведенные до автоматизма навыки работы с циклическими алгоритмами позволяют с легкостью освоить процедуры инициализации, вывода и работы с массивами. Трудности возникают у ряда учеников при выводе результата решения задачи. В этом случае автором предложена следующая методическая находка: предлагается разделить задачи работы с массивами на две основные группы - «заменить» и «найти». Задачи первого типа предполагают изменение всех или некоторых элементов матрицы, в результате решения необходимо вывести получившийся массив на экран. Задачи типа «найти» не требуют вывода всей матрицы в ответе.

Поскольку практически в каждой разновозрастной и разноуровневой по базовой подготовке группе детей в учреждениях дополнительного образования на начальном этапе первого года обучения наблюдается барьер общения подростков между собой и с новым для них педагогом, с целью развития коммуникативных и межличностных навыков оптимально приемлема к использованию методика работы в парах.

В экспериментальной группе формирования пар на каждом занятии было организовано по схеме, исключаяющей повторы ранее объеди-

ненных в пары. В парах воспитанники проверяют друг у друга теоретическую подготовку к занятиям и учатся высказывать свою точку зрения об уровне подготовки своего партнера по группе: достаточно ли, на его субъективный взгляд, товарищ подготовился к занятию или разобрался с очередным теоретическим вопросом. Если, по мнению партнеров в паре, подготовка была не достаточно успешной, воспитанники, еще раз обращают внимание на выявленные пробелы в знаниях и прилагают усилия по их восполнению.

Каждая из работающих пар стремилась превзойти своих конкурентов в группе по уровню усвоения знаний и умений.

В процессе данной работы в парах имеет эффект формирование условий для конкуренции (лидерства) субъектов обучения в группе. Педагог дополнительного образования является главным экспертом, он определяет наиболее эффективно взаимодействующие микрогруппы. Как правило, результатом работы в парах является развитие коммуникативного и межлич-

ностного навыка у преобладающего большинства участников учебного процесса.

По итогам обучения учащимся вручается удостоверение о завершении курса. Кроме того, в течение обучения каждый школьник формирует для себя индивидуальное портфолио работ, включающее все проектные работы, созданные им за время обучения. По окончании обучения, портфолио записывается на DVD-диск и красочно оформляется надписью «Мои достижения», что способствует созданию «ситуации успеха» для всех школьников не зависимо от уровня успеваемости в объектно-ориентированном программировании.

Таким образом, полученные результаты опытно-экспериментального исследования на базе учреждений дополнительного образования Оренбургской области позволяют говорить об эффективности изучения учащимися объектно-ориентированных языков программирования, используя методический подход, исключая этап предварительного изучения процедурного программирования.

10.02.2011

Список литературы:

1. Андросова Е.Г. Методические и содержательные аспекты построения курса программирования на основе объектно-ориентированного подхода: (Для физико-математических специальностей педагогических вузов). Автореф. дисс. ... канд. пед. наук. М., 1996.
2. Герасименко С.А. Дополнительное образование школьников в области информатики. Вестник Оренбургского государственного университета, 2004, № 7. С. 49-54
3. Ермилина Е.В. Формирование готовности школьников к научной деятельности в традиционных и инновационных условиях дополнительного образования : Автореф. дисс. ... канд. пед. наук : 13.00.01. – М, 2008.
4. Занозина Г.В. Организация и использование проектной деятельности в сфере дополнительного образования // Фестиваль педагогических идей «Открытый урок» 2006-2007.
5. Иванова Д.С. Совершенствование информационной подготовки будущих учителей физики (на примере курса информатики «Основы объектно-ориентированного программирования педагогических приложений по физике»). Автореф. дис. ... канд. пед. наук. – Москва. 2004.
6. Петров А.Н. Совершенствование методики обучения объектно-ориентированному программированию на основе объектно-ориентированного проектирования (на примере дисциплины «Программирование» для будущих учителей информатики). Дисс. ... канд. пед. наук. М. 2009.
7. Петрова Ю.А. Дифференцированный подход при обучении объектно-ориентированному программированию в старшей школе. Петрова Ю.А. Дисс. ... к.п.н. Санкт-Петербург. 2002.
8. Петухов А.Ю. Формирование информационной компетентности школьников в системе дополнительного образования на примере учебного модуля курса «Программирование». Петухов А.Ю. Дисс. ... к.п.н. Бийск. 2006.
9. Щетинский Ю.А. Управление развитием профильной школы информатики в учреждении дополнительного образования детей : Автореф. дисс. ... канд. пед. наук. Магнитогорск. – 2005.

Сведения об авторе: **Нефедова Виктория Юрьевна**, старший преподаватель кафедры информатики и методики преподавания информатики Оренбургского государственного педагогического университета, 460000, Оренбург, ул. Советская, 19, (3532) 771498, e-mail: victory2410@yandex.ru

UDC 004.43(075.3)

Nefedova V. Yu.

METHODOLOGICAL APPROACHES TO THE OBJECT-ORIENTED PROGRAMMING TEACHING IN THE SYSTEM OF SUPPLEMENTARY EDUCATION OF CHILDREN (ON THE BASIS OF THE COURSE FOR THE 8-10TH FORMS)

The article deals with the process of object-oriented programming teaching without previous knowledge of algorithmic structures of structure programming for the pupils of the 8-10th forms in the children's supplementary education system.

Key words: object-oriented programming teaching, children's supplementary education.