

М. И. Жадан, Е. М. Березовская

Факультет математики и технологий программирования,
кафедра вычислительной математики и программирования

ПРОГРАММИРОВАНИЕ КАК ЭЛЕМЕНТ ВОСПИТАТЕЛЬНОГО ПРОЦЕССА РАЗРАБОТЧИКА

Воспитание представляет собой целенаправленный процесс формирования духовно-нравственной и эмоционально-ценностной сферы личности обучающегося, в котором достижение положительных результатов обеспечивается усилиями обеих сторон, всех субъектов образовательного процесса – педагогов и воспитанников. Целью воспитания является формирование разносторонне развитой, нравственно зрелой, творческой личности обучающегося.

Особую роль в системе воспитания студенческой молодежи играют информационные технологии, повсеместное использование которых сформирует новую сферу общественных отношений. Информационные технологии оказывают воздействие на самые различные стороны социального и экономического развития общества, государственного управления, повседневной жизни людей. Они составляют основу процессов информатизации и движения к информационному обществу. Поэтому главная задача развития системы образования – широкомасштабное использование информационных технологий на всех уровнях обучения и воспитания студентов. При этом необходимо учитывать отрицательные тенденции этого процесса. Социальные сети стали важной составляющей жизни современной молодежи. Они нередко заменяют живое общение и выполняют функции ценностного воспитания. В связи с этим современному педагогу в условиях информационной коммуникативной среды необходимо эффективно использовать сетевое пространство в воспитательных целях, более активно использовать сервисы социальных сетей для создания групп по интересам, направлениям деятельности и освещения их работы.

Современные достижения науки и широкое внедрение научных технологий в производственные процессы всех отраслей кардинально изменили не только условия трудового процесса, но и высоко подняли планку требований к выпускникам высших учебных заведений, предъявляемых на рынке труда. Поэтому система высшего образования призвана готовить молодых специалистов с высоким уровнем теоретической и практической подготовки, готовых внедрять новые технологии в практическую деятельность.

Основная задача высшего образования заключается в формировании творческой личности специалиста, способного к саморазвитию,

самообразованию, инновационной деятельности. Решение этой задачи вряд ли возможно только путем передачи знаний в готовом виде от преподавателя к студенту. Необходимо перевести студента из пассивного потребителя знаний в активного их творца, умеющего сформулировать проблему, проанализировать пути ее решения, найти оптимальный результат и доказать его правильность.

В рамках сказанного отметим, как программирование может оказывать влияние на воспитательный процесс разработчика.

Программы в основном обрабатывают числовую или текстовую информацию, реже – графическую. Они хранятся долго и, как правило, предназначены для пользователя. Поэтому их реализация должна быть такой, чтобы они могли «читаться» в дальнейшем.

Посмотрим, какого специалиста получим, если решать, например, некоторые задачи из школьного курса «Информатика» на примере учебника «Информатика-8» и соответствующей рабочей тетради.

Рассмотрим примеры, которые говорят о практически полной не читаемости результатов приведенных в книгах программ. Приведенные протоколы работы программ не дают никакой возможности определить, какая все же задача была запрограммирована.

Пример 1 [1, с. 16]. Составить программу, которая выводит на экран количество десятков и количество единиц в данном числе, если оно двухзначное. В противном случае вывести «Не могу найти цифры».

Программа, реализующая задачу, может быть следующей [1, с. 16]:

```
Program cifry;
Var a,c1,c2: integer;
Begin
  Write('Введите целое число');
  Readln(a);
  If (a>9) and (a<100) then
    Begin
      C1:=a div 10;           {Количество десятков}
      C2:=a mod 10;         {Количество единиц}
      Writeln(c1,' – Количество десятков');
      Writeln(c1,' – Количество единиц');
    End
    else write('Не могу найти цифры');
```

End.

Результаты выполнения программы для чисел 27 и 347:

Введите целое число: 27

2 – количество десятков

7 – количество единиц

Введите целое число: 345

Не могу найти числа

Так как, программа пользователю предъявляется в ехе-файле, то тематику задания предугадать достаточно сложно. Избежать этого могла бы замена текста «Не могу найти цифры» на, например, «Число должно быть двухзначным».

Пример 2 [1, с. 29]. Составить программу, которая находит сумму десяти первых натуральных чисел и вывести результат на экран.

```
Program summa;  
Var i,sum: integer;  
Begin  
  Sum:=0;           {Начальное значение суммы}  
  For i:=1 to 10 do sum:=sum+i;  
  Writeln('Sum = ',sum); {Вывод результата}  
End.
```

Результат выполнения программы:

Sum= 55

Здесь опять последняя запись не дает возможности понять условие решаемой задачи. Если же в операторе вывода результата вместо 'Sum = ' записать 'Сумма первых 10 натуральных чисел = ', то задача определяется однозначно.

Теперь приведем несколько примеров на неудачную читаемость протокола работы программы (результата), по которому невозможно восстановить условие задачи. В рабочей тетради к учебнику «Информатика-8» предлагаются следующие задания.

Пример 1. [2, с.18] Составить программу, определяющую, могут ли три числа a , b , c являться длинами сторон треугольника. Числа a , b , c ввести с клавиатуры.

Пример ввода:

3 10 2

Пример вывода:

Не могут

Пример 2. [2, с. 18] Составить программу, определяющую номер координатной четверти, в которой находится точка с координатами (x, y) . Числа x , y введите с клавиатуры.

Пример ввода:

-5 7

Пример вывода:

2

Пример 3. [2, с. 19] Введите два числа a , b . Выведите на экран номер меньшего из них.

Пример ввода:

7 5

Пример вывода:

2

В предлагаемых примерах информация по вводу/выводу вызывает удивление, хотя мы говорим о воспитании программиста.

В книгах имеются также хорошо оформленные программы и результаты с точки зрения читаемости, однако мы остановились на имеющихся, на наш взгляд, неприятных моментах.

Легко читаемая программа позволяет читателю ее понять и, причем, достаточно быстро. Ее наполняют нужной полезной избыточностью, например, «полезными» комментариями. Общей чертой всех комментариев является то, что они должны быть составными частями программ, задуманными и построенными одновременно с этими программами. Только так можно избежать бесполезных и противоречивых комментариев.

Студент (автор) обычно спешит предъявить результат правильно работающей, по его мнению, программы, что не всегда его характеризует как грамотного и опытного специалиста, уважающего труд другого человека (преподавателя, заказчика). В ходе разработки программ ему нужно прививать усидчивость, эффективность реализации, понятность, самостоятельность и уважение, формировать у студентов необходимые профессиональные и гражданские качества.

Список используемой литературы

1 Миняйлава, Е. Л. Информатика: учеб. пособ. для 8-го кл. общеобразоват. учреждений / Е. Л. Миняйлова [и др.]. – Минск: Нар. асвета, 2010. – 189 с.

2 Овчинникова, Л. Г. Информатика 6 рабочая тетрадь для 8 класса / Л. Г. Овчинникова. – Минск: Аверсэв, 2010. – 122 с.