

УДК 372.8

**ВОЗМОЖНОСТИ РЕАЛИЗАЦИИ ДЕЯТЕЛЬНОСТНОГО ПОДХОДА  
ПРИ ПОДГОТОВКЕ БАКАЛАВРОВ ПЕДАГОГИЧЕСКИХ  
СПЕЦИАЛЬНОСТЕЙ ПРОФИЛЯ «ИНФОРМАТИКА»**

**Акимова И.В., Губанова О.М.**

*Пензенский государственный университет, Пенза, e-mail: ulrih@list.ru*

В данной статье рассматривается возможность применения деятельностного подхода в обучении при подготовке бакалавров педагогических специальностей по предмету «Программирование», в частности – решению задач. Автор выделяет основные действия, которые необходимо отработать при решении задач по программированию, приводит необходимые примеры на каждое действие.

**Ключевые слова:** действие, программирование, задача, упражнение

**REALIZATION OF ACTIVITY APPROACH IN PREPARATION OF BACHELORS  
OF PEDAGOGICAL SPECIALTIES OF PROFILE «COMPUTER SCIENCE»**

**Akimova I.V., Gubanova O.M.**

*Penza State University, Penza, e-mail: ulrih@list.ru*

In this article is shown the possibility of application of activity approach in training by preparation of bachelors of pedagogical specialties in the subject «Programming», in particular – to the solution of tasks. The author allocates the main actions which need to be fulfilled at the solution of tasks on programming, shows many essential examples about every action.

**Keywords:** action, programming, task, exercise

Данный период развития общества является периодом информатизации, и именно информатизация образования находится в авангарде этих изменений общественного уклада, ее цель – подготовить все остальные отрасли к новой парадигме существования. Решающая роль в осуществлении информатизации образования отводится учителю, в первую очередь, информатики. Поэтому объектом нашей научной работы является улучшение процесса подготовки бакалавров педагогических специальностей профиля «Информатика».

С одной стороны, будущий учитель не должен являться профессиональным программистом, а лишь обладать определенными компетенциями по составлению алгоритмов и написанию программ на одном из современных языков программирования. Но с другой стороны, ЕГЭ по информатике и ИКТ предполагает решение задач на программирование и в части А (А12), и в части В (В2, В5, В8, В14), и тем более в части С (С1, С2, С4 из демонстрационного варианта ЕГЭ соответственно). Поэтому задача будущего учителя – бакалавра педагогической специальности – не только самому уверенно решать подобные задания, но и владеть основными приемами составления алгоритма достаточно сложных структурированных программ, использующих составные типы данных и дополнительные алгоритмы.

С точки зрения деятельностного подхода к обучению, ядром и существом учебной деятельности является как раз решение учебных задач, в том числе и при обучении элементам программирования на уроках информатики и ИКТ. Решение задач является тем механизмом, через который осуществляется деятельность, происходит формирование умений и навыков выполнять практические действия. В этом случае фактические знания станут следствием работы над задачами, организованными в целесообразную и эффективную систему.

Рассмотрим основные этапы решения задачи на ЭВМ:

1. Постановка задачи. На данном этапе описываются исходные данные, их свойства, взаимосвязи, а также ожидаемый результат.

2. Построение математической модели решения задачи. То есть запись тех условий, которые даны в задаче, ее решения с помощью математических формул.

3. Составление алгоритма решения задачи. Запись алгоритма может быть выполнена любым способом, понятным исполнителю, например, с помощью блок-схемы, в текстовой форме и т.д.

4. Составление программы на основе алгоритма. При составлении программы алгоритм решения задачи переводится на конкретный язык программирования.

5. Отладка программы. Отладка заключается в поиске и устранении синтаксических и логических ошибок в программе.

6. Тестирование программы и анализ решения с доказательством правильности решения. Необходимо запускать программу на выполнение многократно – при различных значениях начальных условий. Анализ результатов служит подтверждением (опровержением) всех предыдущих этапов. После данного этапа, возможно, потребуется пересмотр самого подхода к решению задачи и возврат к первому этапу для повторного выполнения всех этапов с учетом приобретенного опыта.

Итак, мы выделили несколько составляющих действий. Теперь приведем примеры задач на основе вышеизложенного.

**Задача № 1.** Ввести 2 произвольных числа. Найти их сумму, разность, произведение.

**1. Постановка задачи.** Результат выполнения данного действия – это умение выделять, что дано в задаче, а что необходимо найти.

Преподаватель: Что дано в данной задаче?

Учащиеся: Два числа.

Преподаватель: Это какие-то конкретные числа?

Учащиеся: Нет, произвольные.

Преподаватель: Так что же дано в данной задаче?

Учащиеся: Два произвольных числа.

Преподаватель: А что нужно найти?

Учащиеся: Сумму, разность и произведение этих чисел.

Преподаватель: Верно.

**2. Построение математической модели.** Большая часть задач, решаемых в том или ином языке программирования, имеет математическую подоплеку. Поэтому поиск «математики в информатике» неотъемлемый этап решения большинства задач. Рассмотреть математическую формулу, стоящую за данной задачей – вот результат выполнения этого действия.

Выделяем с учащимися исходные данные, результат и основные формулы.

Исходные данные:

$a, b$  – произвольные числа

Результат:

$S$  – сумма

$R$  – разность

$P$  – произведение

Формулы:

$$S = a + b$$

$$R = a - b$$

$$P = a * b$$

**3. Составление алгоритма решения.** Составление алгоритма решения одно из центральных и требующих особого внимания действий. Предлагаем учащимся составить алгоритм решения данной задачи в виде блок-схемы (рис. 1).

**4. Составление программы на основе алгоритма.** Это действие предполагает переход от алгоритма к записи программы на выбранном языке программирования.

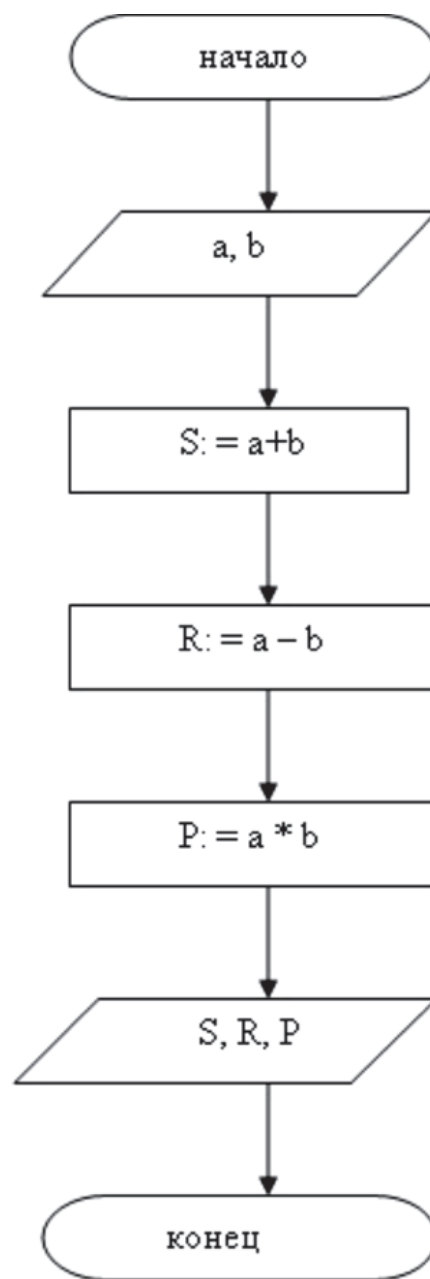


Рис. 1. Алгоритм решения задачи

Program chisla;  
Uses crt;  
Var  
 $a, b, S, R, P$ : real;

```
BEGIN
Clrscr;
Writeln('Ввести два числа');
Readln(a, b);
S: = a + b;
R: = a - b;
P: = a*b;
Writeln('Сумма равна', S);
Writeln('Разность равна', R);
Writeln('Произведение равно', P);
Readln;
END.
```

**5. Отладка программы.** Включает в себя действия по исправлению синтаксических и логических ошибок.

**6. Тестирование программы и анализ результатов.** Предлагаем учащимся проверить работу программы для различных значений  $a$  и  $b$ .

Рассмотрим еще пример.

**Задача № 2.** Найти наибольшее из двух целых чисел, введенных с клавиатуры.

**1. Постановка задачи.** Необходимо проанализировать условие задачи, выявить, что дано, что надо найти.

Учитель: Что дано в данной задаче?

Ученики: две целых числа.

Учитель: Они заданы как константы?

Ученики: Нет, их надо вводить с клавиатуры при запуске программы.

Учитель: Назовите целые типы языка Паскаль.

Ученики: Это, например, типы word, integer, byte.

Учитель: А что необходимо найти в данной задаче?

Ученики: Наибольшее из введенных чисел, максимум из двух чисел.

**2. Построение математической модели.** Математическая модель решения данной задачи достаточно проста. Поэтому учитель может предложить учащимся самостоятельно продумать математическую модель. Возможный вариант: сравнение чисел, сравнение разности чисел с нулем и т.д.

**3. Составление алгоритма решения.** При составлении алгоритма решения, учитель может задать учащимся дополнительные вопросы, касающиеся разветвляющихся алгоритмов. Например, дать определение разветвляющемуся алгоритму, какие виды условных алгоритмов можно предложить, как реализуется ветвление в языке Паскаль.

Результатом такой беседы может стать часть алгоритма решения задачи, представленного в виде блок-схемы (рис. 2).

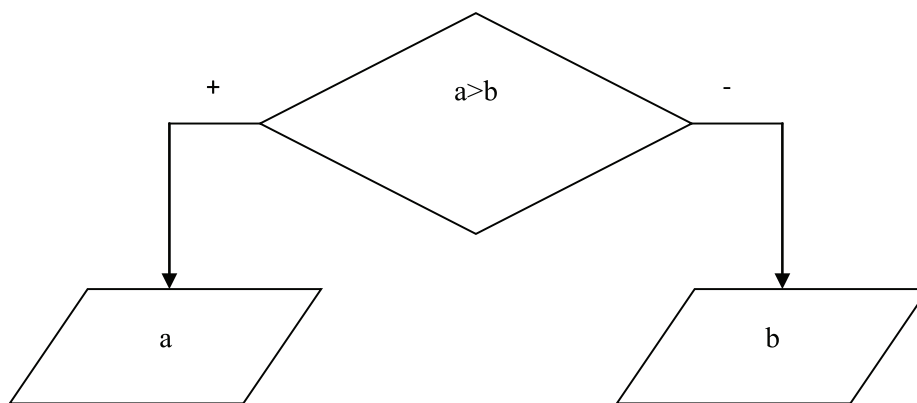


Рис. 2. Фрагмент алгоритма решения задачи

Полностью алгоритм решения представлен ниже (рис. 3).

**4. Составление программы на основе алгоритма.** Программа на языке Паскаль может выглядеть следующим образом.

```
Program chisla;
Var a,b: real;
```

```
Begin
Readln(a,b);
If a > b then writeln(a) else writeln(b);
End.
```

**5. Отладка программы.** Включает в себя действия по исправлению синтаксических и логических ошибок.

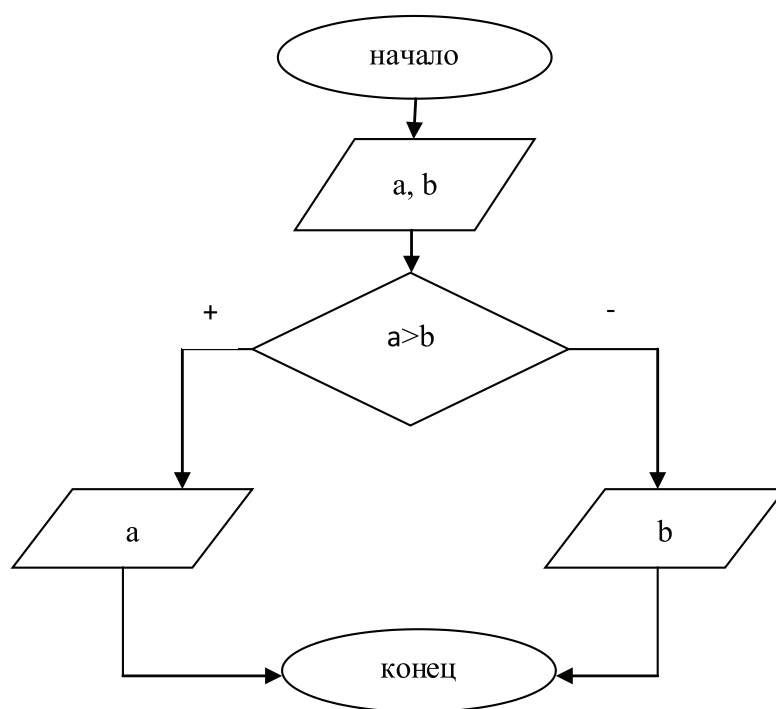


Рис. 3. Алгоритм решения задачи

**6. Тестирование программы и анализ результатов.** Предлагаем учащимся проверить работу программы для различных значений  $a$  и  $b$ .

На каждое из выделенных действий может быть составлена система упражнений для их отработки.