

М.А. Ермаганбетова, Л.С. Сармантаева
РОЛЬ ЗАДАЧИ В ФОРМИРОВАНИИ НАВЫКОВ ПРОГРАММИРОВАНИЯ

Madinaerm111@rambler.ru

Евразийский национальный университет им.Л.Н.Гумилева, Астана

In this article is considered the role of tasks in formation of skills of programming. Some classes of the tasks are selected which decision allows to organize effectively process of training in programming for pupils of high schools.

Программирование является одним из традиционных направлений науки информатики, т.е. это раздел информатики, задача которого – разработка программного обеспечения компьютерной техники. В узком смысле слово «программирование» обозначает процесс разработки программных средств на конкретном языке программирования.

Существуют различные парадигмы программирования, при этом преподавание каждой из них имеет свои методические особенности. Парадигмой программирования подразумевает исходную концептуальную схему постановки проблем и их решения является инструментом грамматического описания фактов, событий, явлений и процессов, условно объединяемых в общее понятие. Каждая парадигма программирования имеет свой круг программистов и класс успешно решаемых ими задач. К основным парадигмам программирования относятся следующие:

1. процедурное программирование (Паскаль, СИ, Бейсик, Фортран, Ассемблеры и т.д.);
2. логическое программирование (Пролог);
3. функциональное программирование (Лисп);
4. объектно-ориентированное программирование.

Классической, более универсальной и наиболее распространенной является процедурная парадигма. Наибольшее количество существующих языков программирования относятся к этой парадигме. В связи с этим чаще всего для обучения программированию в учебных заведениях используется процедурное программирование.

Опыт показывает, в процессе обучения программированию немаловажную роль играет задачи. Обучение программированию должно проводиться на примерах типовых задач с постепенным усложнением структуры алгоритмов решения задач. Кроме этого, для изучения самого языка программирования, т.е. для ввода новых средств, понятий языка опять же требуются решения очередного типа задач.

По значению, обучающие задачи, т.е. задачи, которые с их помощью можно обучать основным приемам программирования, делятся на следующие группы:

1. эвристические задачи, т.е. задачи для ввода новых понятий, средств программирования;
2. задачи для закрепления навыков программирования;
3. задачи для формирования стиля программирования;
4. задачи для развития стиля творческого мышления и т.п.

В свою очередь обучающих задач можно сгруппировать по тематике, например, задачи на вычисления, задачи на обработку текстовых информации, задачи на графику, задачи программированию звука, задачи сортировки и т. п.

Для примера будем рассматривать некоторые классы задач, изучение которых будет необходимо для дальнейшего обучения программированию.

1. Задачи на операции ввода и вывода. Подобный класс задач важен, так как, выполняя подобные задания, учащиеся смогут понять формальные требования, которые предъявляются к программам при их автоматическом тестировании.

2. Задачи на оператор присваивания. Несмотря на кажущуюся простоту, в данный раздел могут быть включены задачи различной сложности. В частности, требование не использовать в программе условный оператор и операторы цикла переводит ряд элементарных задач в задачи повышенной сложности. Такое ограничение с методической точки зрения можно рассматривать как работу с исполнителем, имеющим ограниченный набор команд.

Пример. Ввести с клавиатуры 2 целых числа: $0 \leq N < 12$, $0 \leq M < 60$, указывающие момент времени N часов M минут. Определить и вывести на экран наименьшее число полных минут, которое должно пройти до того момента, когда часовая и минутная стрелки на циферблате совпадут.

При решении данного класса задач учащиеся осваивают основные операции и функции изучаемого языка программирования. Особое внимание при этом уделяется арифметике остатков, знания по которой, как показывает практика, у учащихся недостаточны. Однако именно операции целочисленной арифметики играют важную роль в программировании.

3. Задачи на логические выражения.

4. Условный оператор. Цель данного раздела — научить учащихся составлять алгоритмы с использованием простых и вложенных операторов условия и выбора. Обратите внимание на последовательность – сначала логические выражения, потом условный оператор.

5. Операторы цикла. Основные классы задач, относящиеся к данному разделу, это задачи на обработку числовых последовательностей без использования массивов для их хранения, включая нахождение НОД или НОК для последовательностей натуральных чисел, задачи на рекуррентные соотношения и суммирование числовых рядов. В последнем случае важное значение имеет эффективность программы. Так, при суммировании рядов, аналогичных рядам Тейлора или рядам Маклорена, операции возведения в степень или циклы для вычисления целочисленной степени или факториала применяться не должны. Вместо этого каждое следующее слагаемое должно выражаться через предыдущее с помощью основных арифметических операций.

6. Строковые величины. Цель данного раздела – ознакомить учащихся со строковым типом данных и показать приемы обработки информации такого типа при помощи базовых операций: конкатенации, «вырезка» справа, слева и с любой позиции строки.

7. Простейшие операции над одномерными массивами. Цель данного раздела – научить учащихся правильно описывать, вводить, выводить и обрабатывать одномерные массивы. Стандартными в данном разделе следует считать задачи на поиск максимального и минимального элементов, сдвиг и перестановку элементов массива и, наконец, его сортировку. Следует использовать также нетривиальные задачи, при решении которых используются одномерные массивы, например, вычислить по схеме Горнера значение полинома или такие.

Пример. С клавиатуры вводятся два целых числа $0 < m, n < 101$, а затем $m + n$ элементов целочисленных элементов массива, каждый из которых по модулю не превосходит 32 767. Не

используя дополнительных массивов, требуется переставить местами первые n и последующие m элементов массива. Вывести на экран полученный в результате перестановки массив, разделяя его элементы пробелами.

8. Двумерные массивы. Набор задач – тот же, что и для одномерных массивов. Обработку двумерных массивов следует рассматривать как усложненный алгоритмический (не математический!) вариант простых массивов. Согласитесь, что основным усложнением будет правильное использование вложенных циклов для решения задач с использованием двумерных массивов.

9. Процедуры и функции. Наиболее удачным примером для применения процедур на начальном этапе обучения программированию являются задачи сортировки и поиска. Другой класс задач, требующий естественного применения процедур и функций, – простейшие численные методы интегрирования, а также решение трансцендентных уравнений методами деления пополам, хорд, касательных или итераций.

Дополнительными к перечисленным категориям являются программирование графики и программирование операций с файлами.

Таким образом, сформулированы лишь некоторые из тем, задания по которым могут быть полезны для широкого круга учащихся для формирования основных навыков программирования.

Библиографический список

1. Лапчик М.П., Семакин И.Г., Хеннер Е.К.. Методика преподавания информатики. Учебное пособие для студ. Пед. Вузов.- 2-е издание, стер. – М.: Издательский центр «Академия», 2005.- 624 с.
2. Гусева А. И. Учимся информатике: задачи и методы их обучения. М.: «Диалог МИФИ», 1999 – 320 с.

О.В. Жевняк, Л.И. Филющенко, Е.Г. Шаблова, Т.П. Шишулина ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ПРИ ПРЕПОДАВАНИИ ЮРИДИЧЕСКИХ ДИСЦИПЛИН В УРАЛЬСКОМ ФЕДЕРАЛЬНОМ УНИВЕРСИТЕТЕ

pred@ustu.ru, zevnyak@mail.ru

*Уральский федеральный университет имени первого Президента России Б.Н. Ельцина,
Екатеринбург*

The paper considers an application of information technologies in juridical teaching in Urals Federal University

Преподавание юридических дисциплин должно соотноситься с современными тенденциями развития общества и образования. Принят новый Федеральный закон от 29.12.12 г. № 273-ФЗ «Об образовании в Российской Федерации». Закон уделяется особое внимание использованию информационных, в том числе новых технологий в образовании. В указанном законе подчеркивается значимость сетевой формы реализации образовательных программ (ст. 15), а также реализации образовательных программ с применением электронного обучения и дистанционных образовательных технологий (ст. 16). Применение таких технологий способствует осуществлению процесса обучения независимо от места нахождения обучающихся. Эти технологии актуальны и при преподавании юридических дисциплин.

В Уральском федеральном университете в образовательном процессе используются различные информационные технологии: ресурсы зональной университетской библиотеки, в