

## К ВОПРОСУ СТАРТОВОГО ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Сандро Барнабишвили, Теодор Заркуа

Грузинский университет им. Святого Андрея Первозванного Патриаршества Грузии

### Аннотация

*Описывается методика стартового обучения программированию, основанная на использовании модификации машины Тьюринга. Сформулированы требования к соответствующему программному обеспечению. Приводятся примеры, иллюстрирующие использование предлагаемой методики. Обосновываются преимущества рассматриваемой методики. Излагается взгляд на возможное развитие данной методики.*

### Abstract

*This paper is intended to describe a concept of learning computer programming basics based on a modified version of the Turing machine. Main requirements to the corresponding software are formulated. The publication contains examples illustrating the use of the given approach. There are justified the benefits of the described methodology. Finally, considerations about the further development of the concept are stated.*

### Введение

Традиционная схема создания программного обеспечения, когда от программиста не требуется знания предметной области, уже не всегда может обеспечить требуемое качество продукта. Отсюда, знание основ программирования часто становится необходимо специалистам предметных областей, далеких от информатики. С другой стороны, масса молодых людей с юных лет привыкла иметь дело с компьютером, но не зная основ программирования, использует компьютер исключительно в развлекательных целях и этим самым наносит вред своему здоровью, бюджету и ресурсу времени. Часто это становится причиной серьезных неприятностей во вполне благополучных семьях.

В сложившейся ситуации разработка эффективной методики обучения программированию становится архиактуальной. Данное утверждение усиливается вследствие прекращения поддержки развития систем программирования, основанных на Паскале (методики обучения на базе языка Паскаль были де факто стандартом).

### Требования к методике стартового обучения программированию

Методика обучения программированию должна позволять использовать компьютер как можно раньше и эффективнее. Предлагаемая концепция стартового обучения программированию на базе модификации машины Тьюринга, отталкивается от следующих требований:

1. Сохранить основные части машины Тьюринга – информационную ленту, головку считывания–записи, а также эквивалент таблицы состояний;
2. Возложить функции таблицы состояний на программу, написанную в терминах системы команд модифицированной машины Тьюринга;
3. Сохранить сочетание минимума средств программирования с соблюдением условия универсальности при отсутствии ограничения на ресурсы;
4. Обеспечить простоту и максимальную наглядность в учебном процессе.

### Описание модифицированной машины Тьюринга и примеры ее использования

Предлагаемая модификация машины Тьюринга, состоит из линейной ленты, разделенной на однородные ячейки (естественно, конечной), головки считывания-записи, в каждый данный момент указывающей на текущую (актуальную) позицию, и управляется посредством всего 6 команд. Вот эти команды:

1. L – сдвиг головки на одну позицию влево;
2. R – сдвиг головки на одну позицию вправо;
3. S – прекращение выполнения программы;
4. W<c> – запись символа <c> в текущую позицию ленты;
5. G<m> – безусловный переход в программе на метку <m>;
6. I<c><m> – переход в программе на метку <m> при условии, что в текущей позиции символ <c>.

При этом, каждая команда занимает отдельную строчку, а в качестве метки можно использовать любое неотрицательное целое число, не превосходящее 99999, причем метка в программе также занимает отдельную строку – непосредственно перед помечаемой ею командой.

Очевидно, ознакомление с приведенной системой команд не требует длительного времени (как правило, для этого бывает достаточно 30 минут), после чего уже можно приступить к решению простейших задач. Условие каждой задачи включает описание начального состояния машины, и описание конечного состояния. Условие может включать пояснительные примеры.

**Задача 1.** Переход головки в правый конец последовательности символов, если на старте она находится в области, занятой последовательностью.

**Начальное состояние ленты.** На ленте – любая непустая последовательность символов, не включающая пробелы. Во всех остальных позициях – пробелы.

**Начальное положение головки.** Головка может указывать на любое место вышеупомянутой последовательности.

**Конечное состояние ленты.** Конечное состояние ленты идентично начальному.

**Конечное положение головки.** Головка должна указывать на крайне правый символ беспробельной последовательности символов, расположенной на ленте.

Пример 1.



Пример 2.



Решение может иметь следующий вид:

10	20
R	L
I 20	S
G10	

Для экономии места, программа выведена в два столбца (этот же прием использован в реализации при выводе на экран). Как видим, даже в такой простой задаче мы имеем возможность продемонстрировать ветвление и цикл. Слегка изменив условие, а именно, допустив, что вначале головка может находиться не только на последовательности, но и слева от неё, мы сразу получаем возможность продемонстрировать довольно полезную при обучении ошибку – заикливание во входе бы естественном с точки зрения преемственности решения.

5	I 20
R	G10
I 5	20
10	L
R	S

Начинающему не просто заметить неочевидную ошибку – заикливание, когда на старте головка находится на крайнем правом символе последовательности. Тут же можно предложить красивый выход из положения, организовав т.н. прецикл (аналог музыкального затакта) добавлением в самом начале одного сдвига влево.

Желание иметь возможности, эквивалентные подпрограммам (включая рекурсию) привело к небольшому расширению возможностей трех из 6 команд (в этом направлении есть риск переборщить). А именно, добавление символа R в конце обеих команд перехода (G,I) превратило их в переход с возвратом. А сам возврат был возложен на команду S. Теперь эта команда осуществляет возврат на команду, непосредственно следующую за последней по времени выполнения командой перехода с возвратом, на которую возврата еще не было и завершает программу, если нет ожидающих возвратов.

**Задача 2.** Проверка корректности скобочной последовательности.

**Начальное состояние ленты.** На ленте изначально любая непустая скобочная последовательность состоящая только из круглых скобок и не содержащая пробелы. Все остальные позиции ленты заполнены пробелами.

**Начальное положение головки.** Головка изначально указывает на позицию, примыкающую слева к крайней левой позиции последовательности.

**Конечное состояние ленты.** Конечное состояние ленты идентично начальному.

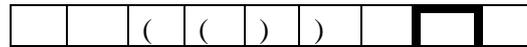
**Финальное положение головки.** Если последовательность корректна, то головка должна быть двумя позициями правее от самого правого символа последовательности, иначе – в любом другом месте.

Пример 1.

Стартовое состояние



Финальное состояние



Решение может быть таким.

10	I(20
R	R
20	S
I(10R	

Как видим, несмотря на лаконичность, решение явно не тривиально. Ту же самую задачу решить без применения расширенных возможностей неизмеримо труднее.

### Заключение

Итак, имеет место одновременно и простота и универсальность (покрываются вопросы, связанные с такими элементами программирования, как ветвление, цикл, рекурсия). Сложность заданий практически неограниченна. Благодаря возможности иметь на экране одновременно как ленту с головкой, так и исполняемую программу, обеспечена наглядность. В дальнейшем планируется, с одной стороны, привести средства простого препроцессинга и этим самым получить возможность раскрыть элементы модульности, с другой стороны, можно развивать программную реализацию созданием арены, позволяющей брать задачи из сети и там же отлаживать их. Автоматизировав учет работы пользователей, можно внести их ранжирование.

Успешно освоивший программирование в рамках предлагаемой методики, в дальнейшем изучает программирование на языке высокого уровня явно в комфортных условиях, соответственно, с большими шансами на успех в обучении.

### Список использованных источников:

1. Пратт Т., Зелковиц М. Языки программирования. Разработка и реализация. СПб., Питер, 2002. – 688 с.
2. Успенский В.А. Машина поста. Москва, «Наука». 1979. – 96 с.