

воспитания необходимо осуществить разработку тематических игровых экологических компьютерных игр разного типа и уровня. Наряду с обучающими играми, знакомящими с редкими видами животных и растений края, региона и республики в охране и знакомящими с правилами поведения в природе (III группа сложности), возможна компьютерная реализация идей экологического образования и воспитания в естественных условиях (экологических тронах), знакомящих с основными элементами экосистем Полесья, их растительным и животным миром (II группа сложности). Перспективными представляются азартные (в хорошем смысле слова) игры соревновательного характера по принципу "экологических полос-ситуаций", в которых играющий попадает в ситуации, когда требуется принимать экологически безопасные решения как для среды, так и для играющего. Конечно, разработка и внедрение пакетов экологических прикладных программ требует определенных средств и усилий, однако реализация их, несомненно, перспективна в плане совершенствования системы экологического образования; даст определенный экономический эффект и будет способствовать созданию благоприятных условий для обеспечения профессиональной подготовки педагога.

А.Ф. Климович, А.И. Павловский

Республика Беларусь, г. Минск,  
Белорусский государственный педагогический университет им. М. Танка

### **НЕКОТОРЫЕ ОСОБЕННОСТИ ПОДГОТОВКИ БУДУЩИХ УЧИТЕЛЕЙ ИНФОРМАТИКИ ПО ТЕМЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ»**

Совокупность функциональных устройств (блоков, элементов) и основных принципов их работы и взаимодействия составляют архитектуру информатики. Она состоит из:

- аппаратных средств (электронных и электронно-механических устройств);
- программных средств (программ, позволяющих организовывать обработку информации);
- алгоритмических средств (средства, на основе которых организовано взаимодействие всех систем ЭВМ).

Аппаратные и программные средства, достаточно подвижная часть информатики. Алгоритмическая часть, на наш взгляд, является основой архитектуры, так как она содержит теоретические основы разработки и

функционирования компьютеров. С помощью языков программирования, в том числе и алгоритмических, создается и программное обеспечение.

В курсе методической подготовки учителя информатики важное место занимают учебные алгоритмические языки (УАЯ) и языки программирования. Но, к сожалению, будущих преподавателей крайне редко знакомят с теоретическими основами разработки вышеупомянутых языков, хотя эти знания являются фундаментальными в области информатики.

Теоретические основы разработки алгоязыков своими корнями уходят в теорию формальных языков. Арто Саломеа отмечает, что потребность в формализованном грамматическом описании различных языков возникает во многих областях наук, и поэтому теория таких языков оказывается междисциплинарным предметом исследования. Альфред Ахо назвал теорию формальных языков «цветом информатики», лепестками которого являются такие разделы, как теория сложности, теория языков программирования и компиляции.

Алгоритмические языки являются искусственными и построены на основе формальных грамматик (точно сформулированных синтаксических и семантических правил). Родоначальником формальных грамматик и языков считается американский лингвист Ноэма Хомский, занимавшийся в 50-х годах исследованиями в области математической лингвистики. Тип грамматик определяется теми ограничениями, которые налагаются на их синтаксические правила. К типу 0 относятся грамматики общего вида (или без ограничений). К типу 1 принадлежат контекстно-зависимые (или не укорачивающие) грамматики. К 2 типу относятся контекстно-свободные (или бесконтекстные) грамматики. К 3 типу – автоматные (размерные) грамматики. Определение типа языка осуществляется в том же порядке, как и определение типа грамматики. Структурированные языки Паскаль, УАЯ КуМир и ИнтАл можно отнести к контекстно-свободным языкам, которые созданы на основе порождающих грамматик.

Для описания языков программирования используют две системы синтаксических обозначений – расширенную форму Бэкуса и синтаксические диаграммы. Эти системы эквивалентны. Но, на наш взгляд, синтаксические диаграммы по своей описательной мощности не уступают металингвистическим формулам и считаются довольно удобными для восприятия, так как дают представление о той или иной структуре языка в целом. По сравнению с метаязыковым описанием синтаксические диаграммы обладают большей наглядностью и доступностью. Так, например, команды учебного алгоритмического языка «если», «пока» и «для» с помощью синтаксических диаграмм будут выглядеть так, как показано на рис. 1.

Кроме самой структуры команды, на диаграммах показана ещё и возможность их вложенности. Синтаксическая диаграмма для команды «для»

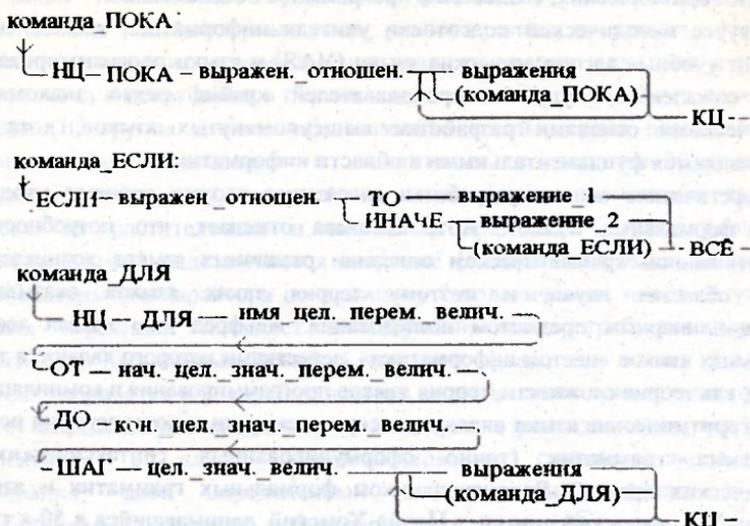


Рис. 1.

также поясняет необходимость использования целочисленного значения величины, по которой организуется цикл. Металингвистическая формула не уточняет эту особенность, что в дальнейшем вызовет ошибки при составлении алгоритмов и соответственно возникают вопросы со стороны учащихся, и как следствие этого увеличиваются затраты учебного времени на разъяснение и исправление алгоритмов во время урока.

Вышесказанное позволяет сделать вывод о том, что основополагающее значение алгоритмических языков в программировании сохраняется и знание теоретических основ их построения, на наш взгляд, являются составной частью информационной культуры учителя информатики. В связи с этим имеет смысл знакомить будущих преподавателей с теорией формальных языков и способами описания конструкций не только в общепринятом виде, но и с помощью синтаксических диаграмм, которые они могли бы использовать в дальнейшей преподавательской деятельности, нагляднее и полнее отражая структуры и формы записи основных понятий и команд учебных алгоритмических языков.