

## **ОПЫТ ПРИМЕНЕНИЯ СРЕДЫ ВИЗУАЛЬНОГО КОНСТРУИРОВАНИЯ ПРОГРАММ НА УРОКАХ ИНФОРМАТИКИ В ШКОЛЕ**

В жизни человека алгоритмы встречаются повсеместно. Рассматривая алгоритм как последовательность действий, можно выяснить причинно-следственные связи и, таким образом, найти средства для достижения искомого результата. Часто именно алгоритм является методикой или технологией созидательного процесса. Несмотря на важность данной темы, в школе ее затрагивают лишь в курсе информатики, да немного на уроках математики.

Что касается информатики, то многие языки программирования требуют однозначного описания последовательности операторов, то есть алгоритма. До недавнего времени программирование составляло основной, а изначально – единственный раздел информатики. Совершенствование технических и программных средств привело к тому, что алгоритмизации и программированию оставляется все меньше места. Современные визуальные, а тем более – мультимедийные пользовательские среды являются теми конкурентами, которые вытесняют разработку программ из сферы интересов школьника. Для работы за компьютером, для поиска информации в сети пользователь имеет простые красивые инструменты, не требующие мыслительных усилий при применении. Между тем, умеющие составить алгоритм и написать по нему программу обычно значительно легче овладевают и пользовательскими навыками, так как понимают механизмы управления компьютером. Эти школьники лучше успевают и по другим предметам, поскольку культура их мышления выше, а ассортимент методов вы-

полнения различных работ богаче. На преподавателя информатики возлагается важная миссия привить учащимся навыки алгоритмического мышления.

Чтобы выполнить эту миссию учитель должен заинтересовать ученика. Сегодня это можно сделать, только активно используя компьютер, причем задание должно быть интересным, инструментальная среда не только дружественной, но и красивой, а результат работы применимым и наглядным на каждом этапе выполнения. Педагогу в настоящее время предлагается выбор между традиционными языками программирования (BASIC, PASCAL, C++) и специальными дидактическими средствами (“Кенгуренок”, “Пылесосик”). Первые ненаглядны и сложны для понимания, вторые – не позволяют получать практически ценный результат.

Учителю нужна среда, дающая возможность, с одной стороны, доступно и наглядно изобразить алгоритм, с другой, – сформировать по этому алгоритму программу на одном из языков, как можно меньше вникая в абстракции языковых конструкций. Получение исполняемого алгоритма (программы) оказывается стимулом для более глубокого освоения предмета.

Разработанная и предложенная для экспериментальной проверки среда визуального конструирования программ BLS [1] отвечает изложенным требованиям. Пакет обеспечивает формирование блок-схемы алгоритма из образных конструкций. Технология разработки позволяет сразу после построения конструкции описать ее назначение на русском языке

и только потом редактировать программный код, исходный шаблон которого предлагается автоматически. Инструментальная среда, стимулирующая такую технологию, позволяет не только повысить эффективность восприятия материала, сократив при этом время работы над темой, но и получить в результате готовые программы, ясные по смыслу и хорошо документированные.

При поиске темы работы автор руководствовался стремлением заинтересовать детей. Для создания алгоритмической схемы была выбрана задача, условно названная “детектор памяти”. Дети шаг за шагом создавали все более сложную модель контроля способности к запоминанию. Усложнение модели требовало знакомства с новыми алгоритмическими конструкциями. Ввод каждой конструкции сопровождался совершенствованием программы. Очередной вариант сразу проверялся. Компьютер активно использовался на всех уроках.

Ниже описан цикл из четырех уроков, реализующих начальную версию “детектора”. Данный цикл предлагался в 7-х и 8-х классах. Практика показала, что учащиеся 8-х классов способны продолжать развитие модели в течение 12-15 уроков, познакомившись в итоге с основными алгоритмическими конструкциями, включая циклические, и оставив на 9-й класс знакомство с массивами. В 7-х классах пришлось ограничиться 8 уроками по заданной теме, подробно разобравшись с ВВОДОМ-ВЫВОДОМ, ПРИСВАИВАНИЕМ, БЛОКОМ РЕШЕНИЕ и лишь затронув ЦИКЛИЧЕСКИЕ алгоритмы. Разрабатывая серию уроков, следует связать длину этой серии с возрастом детей: чем моложе ученики, тем короче должна быть серия. Автору очень помогло то, что каждый урок завершался готовой программой, а каждые три-четыре урока – функционально законченным разделом. Это позволило прервать цикл в 7-х классах, как только интерес к материалу стал ослабевать.

## СХЕМА “ДЕТЕКТОР ПАМЯТИ”. НАЧАЛЬНАЯ ВЕРСИЯ.

### УРОК 1. ПОСТАНОВКА ЗАДАЧИ.

*Цель урока.* Сформулировать задачу, найти алгоритм решения, начать практическое знакомство со средствами реализации.

*Форма проведения.* Образное обсуждение, затем демонстрация среды BLS на компьютере.

#### *Краткий план:*

- Объяснение целей работы 7 мин.
- Живое обсуждение возможных алгоритмов и выработка единого подхода и алгоритма 20–25 мин.
- Работа за компьютером 8-13 мин.

*Содержание обсуждения.* Как проверить память? Наверное, нужно что-то сказать или **показать** и предложить запомнить. Чем **дольше** испытуемый будет смотреть на запоминаемый объект, тем лучше запомнит. Далее, объект должен быть удален, а проверяемому предоставлена возможность **воспроизвести** запомненное. **Сравнение показанного с воспроизведенным** позволяет охарактеризовать качество памяти.

Такова простейшая модель, разработанная совместно с учениками. Она допускает существенное развитие в дальнейшем.

*Результат урока.* Сформулирован алгоритм, предназначенный для компьютерной реализации модели:

1. Вывод на экран текста для его запоминания испытуемым.
2. Задержка на некоторое время, чтобы испытуемый мог прочесть и запомнить текст.
3. Удаление текста с экрана.
4. Ввод испытуемым запомненного текста с клавиатуры.
5. Сравнение введенного с клавиатуры текста с исходным.

6. Вывод на дисплей сообщения об итогах испытания по результату сравнения.

Начата компьютерная реализация: вызвана инструментальная программа и сконструирована схема, показанная на рис. 1.

Данная схема демонстрирует тот важный факт, что любой алгоритм должен начинаться и закончиться, причем начало всегда предшествует концу. Отметим, что для создания этой схемы достаточно вызвать программу BLS и ввести имя. Это имя должно отражать назначение программы, например, DetMem (Detector of Memory) или аналогичное. Полезно сделать вывод о том, что схема записывается в файл. Файл, таким образом, является носителем программы, а его имя – средством нахождения файла. Полученный файл должен быть использован в дальнейшей работе.

*Вопрос для самостоятельного обдумывания.* Что должно находиться в схеме между **началом** и **концом**?

## УРОК 2. НАЧАЛО РЕАЛИЗАЦИИ АЛГОРИТМА.

*Цель урока.* Практическая реализация п.п. 1, 2, 3 алгоритма, полученного на первом уроке. Для этого необходимо познакомиться с алгоритмическими конструкциями “Ввод-вывод” (сначала – “Вывод”), “Процесс”. Научиться основам работы с BLS: вставлять визуальные конструкции, редактировать автоматически генерируемый программный код, записывать комментарии, сохранять результаты своей работы на магнитном носителе (в файле).

*Форма проведения.* Практическая работа за компьютером, чередуемая демонстрацией учителем приемов работы и обсуждением возникающих трудностей.

### *Краткий план:*

- Конструкция вывода, ее графическое представление, текстовая программная реализация, задание индивидуального текста для запоминания испытуемым, комментариев, запуск программы 10 мин.

- Блок “Начало – конец”: корректировка текста программы, комментарий о назначении программы и авторе. Запуск пробного работающего варианта (текст выводится на экран) 8 мин.

- Реализация блока временной задержки текста на экране через процедуру Delay и удаления текста с экрана через процедуру ClrScr. Использование блока “Процесс” 10 мин.

- Варьирование времени задержки текста и самого текста, самостоятельная работа 12 мин.

*Результат урока.* Дети создали схему. Для этого они познакомились с тремя алгоритмическими конструкциями: “Начало-конец”, “Ввод” и “Процесс”. Схема позволяет увидеть графическое представление конструкций и сопоставить этому представлению программный код. Отчетливо видна последовательность выполняемых действий: сверху вниз. Показано также, что каждый блок обязательно будет исполнен, причем только один раз.

Полученная схема показана на рис. 2. Фрагмент действующей программы приведен ниже:

```
----- Pascal-программа -----
(* Программа "Детектор памяти" к уроку 2.
Выполнена учителем Хайтом А.М. *)
PROGRAM MEMORYDET(INPUT,OUTPUT);
USES CRT;
BEGIN
  (* Вывод на экран текста для запоминания *)
  WRITELN('Контрольный текст');
  (* Задержка и очистка экрана *)
  DELAY(20000);
  CLRSCR
END.
```

*Домашнее задание.* Изобразить пройденные на уроке алгоритмические конструкции, написать, какие функции выполняют эти конструкции в решаемой задаче. Описать с помощью изученных конструкций алгоритм посещения школьной столовой.

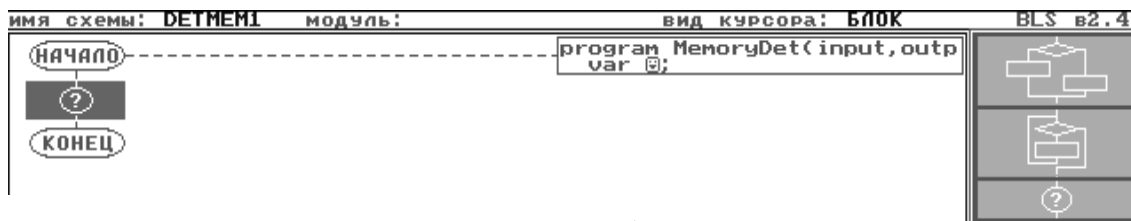


Рисунок 1.

### УРОК 3. ПОНЯТИЕ ПЕРЕМЕННОЙ. ВВОД С КЛАВИАТУРЫ.

*Цель урока.* Практическая реализация п. 4 алгоритма. Для этого, помимо блока “Ввод-вывод” и функции ввода, требуется понятие переменной. Переменная, с одной стороны, место в оперативной памяти, куда будет введен с клавиатуры запомненный испытуемым текст, с другой, – имя и тип, формально описываемые в разделе VAR блока “Начало-конец”.

*Форма проведения.* Объяснение понятия переменной с примерами, поясняющими, что такое тип. Опрос – собеседование по объясненному материалу. Практическая работа за компьютером.

#### Краткий план:

- Объяснение материала 15 мин.
- Опрос по пониманию объясненного и уточнению понятий 10 мин.
- Практическая работа за компьютером 15 мин.

*Результат урока.* Получена новая версия схемы. Проверено, что вводимый

с клавиатуры текст попадает “в компьютер”. Введено важное и сложное понятие переменной, уточнено назначение блоков “Начало-конец”, и “Ввод-вывод” (Ввод).

#### Домашнее задание:

- Описать три различные переменные строчного типа.
- Ответить на вопросы: В каком разделе описываются переменные? Куда попадает информация, вводимая пользователем с клавиатуры?

### УРОК 4. БЛОК ВЕТВЛЕНИЯ ПО УСЛОВИЮ “ЕСЛИ”.

*Цель урока.* Получить законченную версию программы, то есть дополнить схему блоками, выполняющими п.п. 4 и 5 алгоритма. Для реализации п. 4 необходимо познакомиться с условным оператором, графически представленным как блок “Решение”. В отличие от предыдущих конструкций этот блок имеет две ветви. При выполнении п. 5 учащиеся повторяют и закрепляют материал, связанный с конструкцией “Ввод-вывод”.

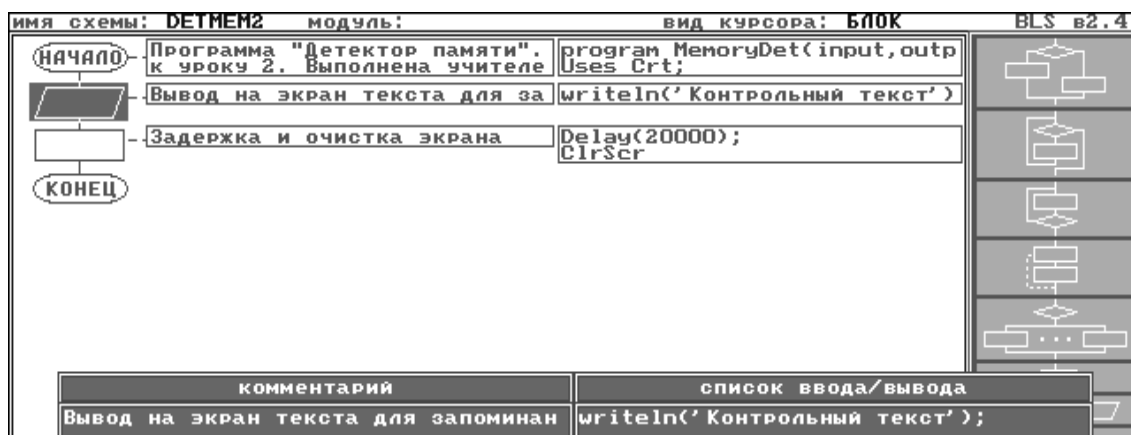


Рисунок 2.

*Форма проведения.* Работа за столом с конспектом, обсуждение темы, практическая работа за компьютером.

*Краткий план:*

- Объяснение материала 12 мин.
- Практическая работа над алгоритмом за компьютером 20 мин.
- Проверка первого варианта функционально законченной программы 8 мин.

*Результат урока.* Получаем первый вариант работающей программы, построенной по приведенному на рис. 3 алгоритму, а также его программную реализацию: (\* Программа "Детектор памяти". Схема к уроку 4. Выполнена учителем Хайтом А.М. \*)  
PROGRAM MEMORYDET(INPUT,OUTPUT);  
USES CRT;

VAR

TestText:String;

BEGIN

(\* Вывод на экран текста для запоминания \*)

WRITELN('Контрольный текст');

(\* Задержка и очистка экрана \*)

DELAY(20000);

CLRSCR;

(\* Ввод испытуемым текста \*)

READLN(TestText);

(\* Сравнение введенного текста с заданным \*)

IF TestText = 'Контрольный текст' THEN

BEGIN

(\* Вывод сообщения при правильном ответе \*)

WRITELN('Память хорошая');

END

ELSE

BEGIN

(\* Вывод сообщения при неправильном ответе \*)

WRITELN('Память нужно тренировать!');

END

END.

Схема показывает, что некоторые блоки останутся невыполненными при определенных условиях, причем для одних условий будет реализована одна ветвь схемы, для других – другая. Смысл блока “Решение” становится очевидным. В завершающей части урока учащиеся смогли “проверить память” одноклассников.

#### ИТОГИ.

За четыре урока дети получили готовый программный продукт, которым можно воспользоваться, показать родителям, товарищам. Сгенерированная программа написана в “хорошем стиле”, то есть каждая ее строка понятна, соответствие между алгоритмическими конструкциями и текстом очевидно.

Учащиеся познакомились с четырьмя алгоритмическими конструкциями и ис-

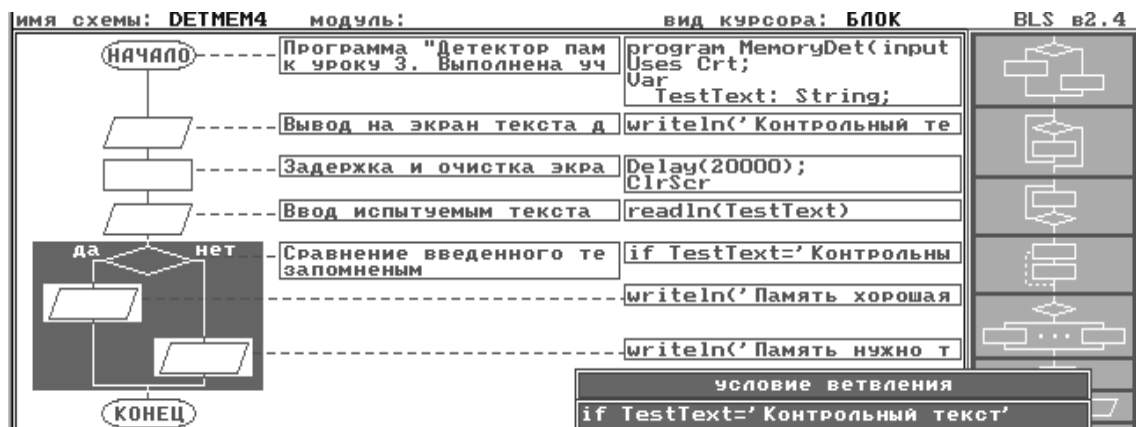


Рисунок 3.

пользовали их на практике, узнали, что такое линейная и разветвленная структуры алгоритма. Наконец, было рассмотрено понятие переменной и также применено на практике. Важно и то, что на всех этапах работы участие детей было активным и творческим, а инструментальная среда делала наглядным каждый шаг создания схемы.

На прилагаемой дискете содержатся файлы со схемами, полученными на каждом уроке.

Для практических работ по курсу “Алгоритмизация и программирование”, учитель может так же использовать примеры, предлагаемые в учебнике Л.З. Шауцковой [2].

### ЗАКЛЮЧЕНИЕ

Предложенные методики позволяют быстро познакомиться с основными алгоритмическими конструкциями, привить навыки грамотной разработки программы. При переходе к более сложному материалу результативность наглядного метода

#### Литература.

1. Смольянинов А.В. Возможности среды визуального программирования BLS для использования в учебном процессе. Наст. выпуск. Стр.
2. Шауцкова Л.З. Информатика: Учебник для 7-11 кл. общеобразоват. учебн. заведений. В 2 кн. Кн. 2: Практика алгоритмизации и программирования. 2-е изд., перераб. и доп. – Нальчик: Эль-Фа, 1997.

*Хайт Александр Моисеевич, учитель информатики гимназии № 56, г. Санкт-Петербург.*



Рисунок 4.

обучения еще выше. Среда BLS универсальна, что позволяет реализовывать с ее помощью как примеры из учебника, так и авторские замыслы педагога.

Возможности BLS не исчерпываются разработкой программ. На рис. 4 приведен алгоритм и представлено краткое содержание данной статьи, выполненное в форме схемы.

НАШИ АВТОРЫ