

5. PascalABC.NET [Электронный ресурс]/ Режим доступа: <http://ru.wikipedia.org/wiki/PascalABC.NET>. Дата обращения: 05.04.2013.
6. Free Pascal [Электронный ресурс]/ Режим доступа: http://ru.wikipedia.org/wiki/Free_Pascal. Дата обращения: 05.04.2013.
7. Embarcadero turbo pascal download [Электронный ресурс]/ Режим доступа: <https://downloads.embarcadero.com/free/tp>. Дата обращения: 05.04.2013.
8. Слинкин, Д. А. Основы программирования на Турбо-Паскале : учеб.-метод. пособие для студентов вузов / Д. А. Слинкин. - Шадринск : ШГПИ, 2003. - 217 с
9. Слинкин, Д. А. Разбор и реализация алгоритмов типовых задач курса "Языки и методы программирования (ТУРБО-ПАСКАЛЬ)" [Текст] : метод. материалы для студентов пед. вузов / Д. А. Слинкин. - Шадринск : ШГПИ, 2003. - 41 с.
10. Ахо А. и др. Структуры данных и алгоритмы / Пер. с англ. - Уч. пос – М.: Издательский дом “Вильямс”, 2000. - 384 с., ил.
11. Вирт Н. Алгоритмы и структуры данных / Пер. с англ. – М.: Невский Диалект, 2001. – 352 с.
12. А.П. Полищук, С.А. Семериков, Системное программирование в UNIX средствами Free Pascal [Электронный ресурс], 2005.– 419с . Режим доступа: <http://freepascal.ru/download/book/sysfpc.zip>. Дата обращения: 05.04.2013.
13. Мансуров К.Т. Основы программирования в среде Lazarus [Электронный ресурс], 2010. – 772 с.: ил. Режим доступа: http://www.freepascal.ru/download/book/lazarus_osnovy/osnovy_programmirovaniya_v_srede_lazarus.pdf. Дата обращения: 05.04.2013.
14. Алексеев Е. Р., Чеснокова О. В., Кучер Т. В. Free Pascal и Lazarus: Учебник по программированию / Е. Р. Алексеев, О. В. Чеснокова, Т. В. Кучер — М. : ALT Linux ; Издательский дом ДМК-пресс, 2010. — 440 с. : ил. — (Библиотека ALT Linux).
15. О.В. Деревенец. Песни о Паскале [Электронный ресурс]/, 2012. – 590с. Режим доступа: http://www.freepascal.ru/download/book/pesni_o_paskale_v12.4.pdf. Дата обращения: 05.04.2013.

**РАЗРАБОТКА СРЕДСТВ ВИЗУАЛИЗАЦИИ АЛГОРИТМОВ
ДЛЯ ПРИМЕНЕНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ШКОЛЬНИКОВ
АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЮ**

П. Н. Туголуков
Научный руководитель: П.И. Алексеевский, ст. преподаватель

Аннотация

Статья посвящена исследованию применения графического изображения алгоритмов (блок-схем): причины появления, стандарты блок-схем, использование в учебном процессе. В данной работе показан ход решения некоторых задач основного исследования: аргументация создания инструмента, совмещающего в себе наглядность блок-схем и возможность формальных языков к выполнению алгоритмов, описание создания инструмента.

Ключевые слова: алгоритм, учебный алгоритмический язык, язык программирования, обучение основам алгоритмизации и программирования, требования к учебным алгоритмическим языкам.

Одной из основных целей курса информатики является развитие у учеников алгоритмического стиля мышления, т.е. способности составить алгоритм для решения поставленной задачи, что в свою очередь требует множества определенных знаний и умений.

В процессе обучения используются различные средства, как практической, так и методической направленности.

Для записи алгоритмов, их демонстрации, анализа используется четыре основных средства:

- неформальный (естественный) язык;
- псевдокод;
- формальный язык (язык программирования);
- графическое представление (блок-схемы).

В качестве инструментов практической направленности, т. е. формирования алгоритмического мышления, используются формальные языки программирования, с помощью алгоритмы записываются для последующего их выполнения и решения поставленных задач. В качестве средств методической направленности, т.е. Средств, направленных на получение основных знаний, используется графическое представление алгоритмов - блок-схемы.

Блок-схема - графическое изображение алгоритма, в котором каждый отдельный шаг представлен отдельным блоком соответствующей формы, соединенных друг с другом линиями (стрелками), указывающих направление хода передачи управления между блоками.

Блок-схемы популярны, так как они графически отображают логику программы с помощью стандартных геометрических фигур и соединительных линий. Они представляют собой «интуитивно понятный метод представления управляющей последовательности алгоритма»(Кушниренко А.Г., Лебедев Г.В. «12 лекций о том, для чего нужен школьный курс информатики и как его преподавать»)

На рис. 1 представлена блок-схема алгоритма игры «Угадайка». Видно, что алгоритм детально описан, структура игры представление достаточно наглядно.

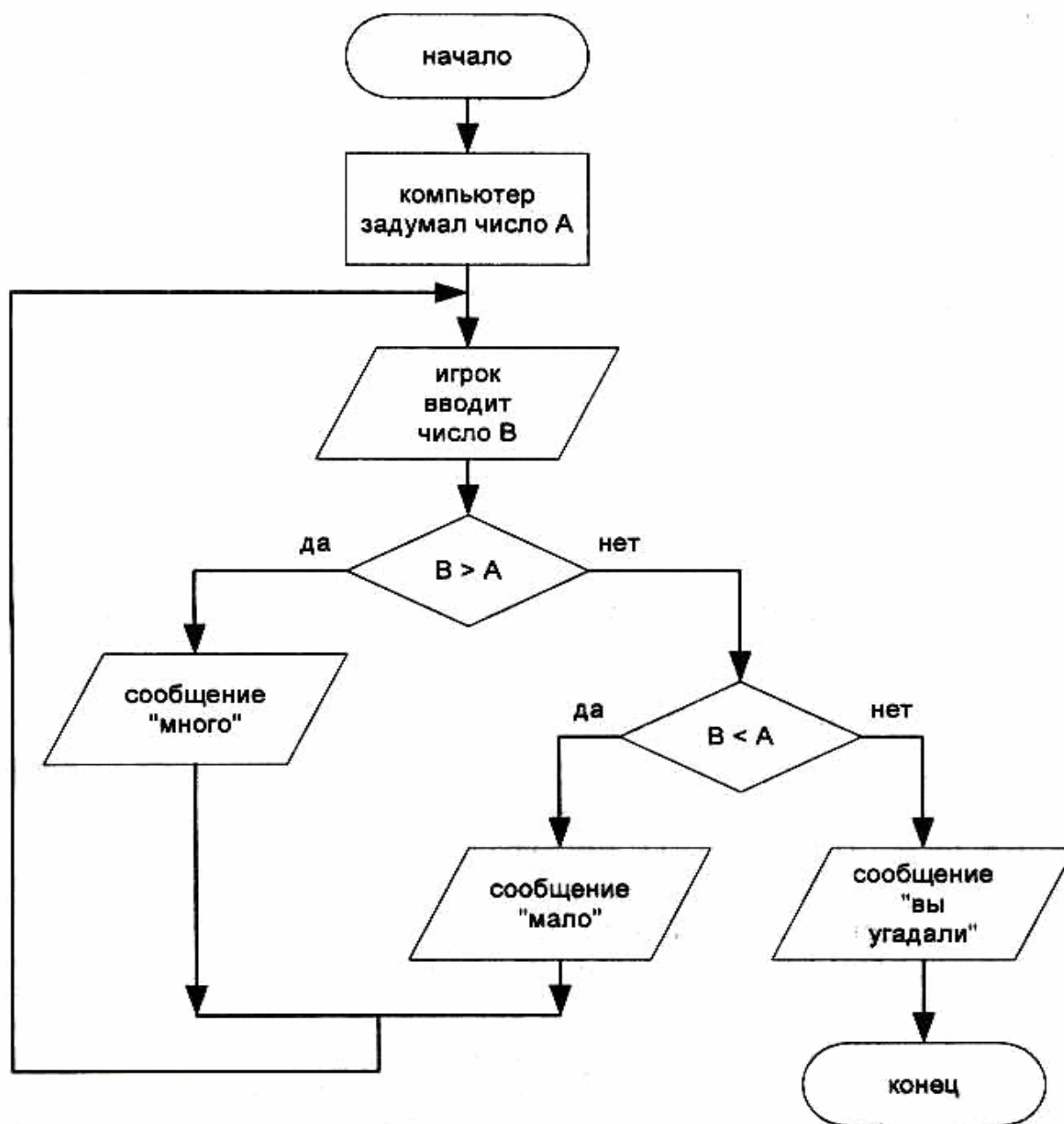


Рис. 1. Блок-схема алгоритма игры "Угадайка"

Блок-схемы, как вид записи алгоритмов, могут соответствовать стандартам. В данной работе за основу взято подмножество стандарта «ГОСТ 19.003-80. Схемы алгоритмов и программ. Обозначения условные графические.» используемое в школах. Блоки:

- Начала алгоритма
- Конца алгоритма
- Процесса
- Условия
- Ввода/вывода

- Цикла с параметром

Так же, в качестве соединительных линий используются стрелочки.

Разграничение использования языков программирования и блок-схем заключается в следующем:

- Пригодность для первоначального изучения. Блок-схемы являются более наглядным средством, нежели формальные языки, потому предпочтение зачастую отдается графической форме записи алгоритмов. Блок-схемы детально раскрывают строение основных алгоритмических конструкций, в то время, как ЯП раскрывают взаимодействие множества отдельных блоков. (Тут следует внести поправку. Психологи учат нас в процессе обучения воздействовать на разные полушария мозга учеников, посредством использования различных форм представления материала. Следуя этому, материал можно преподавать используя и графические наглядные схемы, и абстрактные словесные конструкции. Некоторые ученики почерпнут больше из первого, некоторые из второго, а кто-то — и из блок-схем, и из словесного описания алгоритма).

- Пригодность для практического использования. Блок-схемы, вследствие развернутой структуры блоков имеют больший размер, чем запись аналогичных шагов алгоритма на ЯП, что влияет на размер программы. Это свойство графического представления не дает возможности без особых трудностей создавать большие программы.

Порой трудно перейти от блок-схем к ЯП, поскольку у учеников отсутствует опыт создания выполняемых алгоритмов с помощью графического представления. Таких инструментов в школе попросту нет.

Весомым дополнением к уже существующему инструментарию будет являться инструмент по работе с блок-схемами, который позволит выполнять записанные алгоритмы. Преимущества данного программного продукта:

- Использование для создания программы наглядных графических компонентов, что в свою очередь позволит более полно понять суть основных алгоритмических конструкций.

- Получение выполняемого алгоритма, который можно проверять на работоспособность, исследовать его свойства, анализировать.

- Возможность пошагового выполнения алгоритма (т. н. трассировки).

Разработанный автором инструмент позволит с помощью блок-схем не только записывать алгоритмы, как это было ранее, но выполнять их, анализировать, тестировать, что будет хорошей практикой и опытом для перехода от графического представления алгоритма к представлению с помощью формального языка — языка программирования. Данные функции присущи структурному

программированию, что делает применение такого инструмента еще более обоснованным, поскольку практическое большинство языков программирования, которые изучаются в школах, являются императивными языками, в которых реализована парадигма структурного программирования.

Разработанная в ходе исследования программа является совокупностью трех основных компонентов:

- компонент визуализации,
- компонент компиляции,
- компонент выполнения.

Компонент визуализации — модуль программы, в котором происходит создание графических компонентов, т. е. рисование блок-схем. Для создания данного компонента используется библиотека Qt и язык C++.

Компонент компиляции — модуль программы, отвечающий за трансляцию нарисованных блок-схем в код виртуальной машины. Для создания этого модуля используется утилита bison и GCC.

Компонент выполнения — виртуальная машина, выполняющая код, который она получает от компонента компиляции. Для создания этого компонента использовался язык C и язык Assembler.

Подобная архитектура позволит сделать программу переносимой на различные операционные системы, что на данный момент крайне актуально в связи с тенденцией перехода в школах свободное программное обеспечение.

В результате проделанной работы получен инструмент, с помощью которого можно использовать графическое представление алгоритмов при изучении алгоритмизации и программирования. Функционал программы позволяет выполнять записанные алгоритмы, тем самым делая мягче переход от блок-схем к языкам программирования. Данный инструмент может быть использован в преподавании информатики 5-7 классам при изучении раздела «Основы алгоритмизации и программирования», а так же при изучении других разделов, в которых может возникнуть потребность использования блок-схем.

ЛИТЕРАТУРА:

1. Кушниренко А.Г., Лебедев Г.В. 12 лекций о том, для чего нужен школьный курс информатики и как его преподавать. Лаборатория Базовых Знаний, 2000.
2. Кобилев С.С. Образовательная информатика: Подход к обучению, выбор учебных языков и создание программных систем .
3. Dijkstra E.W. Structured programming,. Academic Press, 1972.