

МИНИСТЕРСТВО ОБРАЗОВАНИЯ САРАТОВСКОЙ ОБЛАСТИ  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ САРАТОВСКОЙ ОБЛАСТИ  
«ПУГАЧЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ЛИЦЕЙ»

**МЕТОДИЧЕСКАЯ РАЗРАБОТКА**  
**УРОКА НА ТЕМУ**  
**«АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»**  
**ПО УЧЕБНОЙ ДИСЦИПЛИНЕ «ИНФОРМАТИКА»**

Составила (разработала) Рябинская М. В., преподаватель информатики ГБПОУ СО «ПЛ».

Рассмотрено на заседании методической комиссии естественно – математического цикла.

Протокол № 1 от 26.08.2020 г.

Председатель МК Мальцева Л. А.

## Методическое обоснование темы.

Курс дисциплины Информатика в учебных учреждениях СПО содержит один из разделов «Алгоритмизация и программирование». В данном разделе рассматривается содержание основных понятий программирования, история его развития, основные элементы и языки программирования. Принципы структурного программирования и начала программирования в среде Turbo и системе Pascal Delphi. Дается материал о программном обеспечении для вычислительной техники и автоматизированных систем.

Программирование можно рассматривать как искусство, науку, ремесло. Программирование – это искусство получения ответов от машины. Для этого в узком смысле нужно составить специальный код для технического устройства, в широком – разработать программы на языках программирования, т.е. не просто составить код, а выполнить интеллектуальную работу по составлению высокоразумных программ для решения различных задач во всех сферах человеческой деятельности.

Тема «Алгоритмизация и программирование» направлена на понимание сути алгоритмов, их свойств, способов описания, так как эта тема развивает: ясность и четкость мышления; способность предельно уточнять предмет мысли; внимательность, аккуратность, обстоятельность, убедительность в суждениях; умение абстрагироваться от конкретного содержания и сосредоточиться на структуре своей мысли.

Методические рекомендации по проведению занятия.

Программирование традиционно относят к сложным темам курса дисциплины «Информатика», признавая при этом, что именно решение задач по теме алгоритмизация и программирование в максимальной степени способствуют развитию алгоритмического стиля мышления, который формирует учебные навыки. Действительно для успешного решения задачи, требующей составления алгоритма и написания программы, учебные занятия надо проводить таким образом, чтобы студент мог:

1. Четко понять задачу, провести ее детализацию и формализацию.
2. Проанализировать, к какому классу задач она относится, какими способами (алгоритмами) ее можно решить.
3. Составить алгоритм решения задачи.
4. Составить программу, реализующую этот алгоритм.
5. Проверить, правильно ли программа работает, ту ли задачу она решает.
6. В случае обнаружения ошибки необходимо проделать все (или некоторые) вышеперечисленные действия заново с целью исправления ошибки.

План занятия:

Дисциплина: Информатика.

Тема программы: «Алгоритмизация и программирование».

Тема занятия: «Алгоритмы и их свойства. Формы записи алгоритмов: словесные, графические».

Тип занятия: изучение нового материала.

Вид занятия: смешанный.

Цели занятия:

- Образовательные: разрешить представление об алгоритмах: ознакомить студентов на примерах со свойствами алгоритмов; объяснить студентам формального алгоритма; дать классификацию типов алгоритмов по структуре их построения. Закрепить полученные знания с помощью электронного теста, развивать навыки самоконтроля.
- Развивающие: развивать познавательный интерес; развивать умение анализировать, обобщать, сравнивать, абстрагироваться, синтезировать знания, полученные на различных дисциплинах.
- Воспитательные: воспитывать стремление к получению новых знаний, обобщению знаний из различных областей жизни; чувство товарищества. Взаимовыручки; коммуникативные качества, умение слушать; культуру межличностных взаимоотношений, аккуратность в работе, трудолюбие.

Материально-техническое обеспечение:

учебный кабинет: посадочные места по количеству обучающихся; рабочее место преподавателя; наглядные пособия; комплект учебно-методической документации.

Технические средства обучения:

компьютеры с лицензионным программным обеспечением; средства мультимедиа, презентация, составленная в программе PowerPoint.

Межпредметные и внутрипредметные связи:

алгоритмическая линия формирует навыки алгоритмического и логического мышления, проектной работы и моделирования. Данная тема способствует развитию алгоритмического мышления, развивает умение читать алгоритмы, умение составить алгоритм для различных жизненных ситуаций и анализировать обстоятельства.

Алгоритмизация, как раздел дисциплины Информатика, который изучает процессы создания алгоритмов, традиционно относится к теоретической информатике вследствие своего фундаментального характера. При этом сторонники «пользовательского» подхода при изучении дисциплины «Информатика», говорит об отсутствии практического значимости этого раздела для развития навыков пользователя современного программного обеспечения. Вследствие развития новых информационных технологий появляется возможность в пределах раздела «Основы алгоритмизации и программирования» давать общенаучные понятия информатики и в то же время формировать и развивать умение и навыки, необходимые пользователю при работе с современным программным обеспечением, т.е. появляется возможность сделать раздел «Основы алгоритмизации» мостиком между теоретической и практической информатикой.

Технологическая карта урока

Этап урока	Задачи этапа	Формы и методы обучения	Планируемый результат
Организационный момент.	Психологически подготовить студентов к работе	Фронтальная, Беседа	Группа психологически подготовлена к работе
Подготовка студентов к активному усвоению знаний.	Организовать и целенаправить познавательную деятельность студентов. Научить студентов формулировать цель	Фронтальная Беседа	Определена проблема, намечены пути ее решения. Поставлена цель.
Усвоение знаний	Создать условия для самоорганизации познавательной деятельности обучающихся	Фронтальная Лекция	Обучающимися изучены основные понятия
Первичная проверка знаний	Самоконтроль усвоения основных понятий	Индивидуальная Самостоятельная работа	Показали знания понятий
Контроль	Контроль глубины понимания изученного материала	Индивидуальная Метод контроля	Показали глубину понимания изученного материала
Рефлексия. Подведение итогов урока.	Мобилизовать учащихся на организацию самооценки полученного	Фронтальная	Поставленные задачи решены, цель достигнута Высказывают собственное мнение о

	результата умения своей деятельности	Развивать самооценки учебной деятельности	том, достигнута ли цель занятия.
Инструктаж о домашнем задании.	Сообщить студентам о домашнем задании	Индивидуальная Работа с лекцией	Творческий характер и правильное выполнение домашнего задания. Дидактический материал к занятию:

Дидактический материал к занятию:

- Алгоритмические задачи (задания на слайдах презентации);
- практикум по решению задач (в программе Pascal);
- электронный тест в тестовой оболочке MyTestEditor.

Список литературы для студентов:

1. *Цветкова М.С.* Информатика и ИКТ: Практикум для профессий и специальностей технического и социально-экономического профилей: учебное пособие для студентов учреждений среднего профессионального образования / под ред. М.С. Цветковой. - М.: 2018

*Малясова С. В., Демьяненко С. В.* Информатика и ИКТ: учебник пособие для студентов учреждений среднего профессионального образования / под ред. М.С. Цветковой. - М.: 2018

*Цветкова М.С.* Информатика и ИКТ: учебник для студентов учреждений среднего профессионального образования – М, 2018

*Цветкова М.С.* Информатика и ИКТ: электронный учебно – методический комплекс для студентов учреждений среднего профессионального образования - М.: 2018

*Цветкова М.С., Хлобыстова И.Ю.* Информатика и ИКТ: Практикум для профессий и специальностей естественно-научного и гуманитарного профилей. - М.: 2018

Ход занятия

1. Организационный момент.
2. Подготовка к изучению нового материала (ознакомление с планом и целью занятия).
3. Изучение нового материала (просмотр электронного урока с использованием мультимедиа проектора: слайды, текст лекции).
4. По ходу занятия студенты конспектируют определения и отвечают на вопросы.
5. Закрепление темы занятия (решение задач, работа студентов на компьютере).
6. Подведение итогов. Выставление оценок с учетом процентного выполнения теста.
7. Задание на дом (выучить определения, привести примеры алгоритмов из жизненной практики).

Каждый человек в повседневной жизни, во время учебы или на работе решает огромное количество задач самой разной сложности. Некоторые задачи просты и привычны, мы решаем их, не задумываясь (собраться на занятия, закрыть дверь, перейти улицу ...). Другие задачи, так трудны, что требуется длительный срок для поиска решения и достижения поставленной цели. Решение даже самой простой задачи обычно осуществляется за несколько последовательных шагов.

Одним из важнейших этапов решения задач на ЭВМ – составление алгоритма. О том, что такое алгоритмы, какими общими свойствами они обладают и как исполняются, мы и поговорим на этом занятии.

Как вы думаете, откуда произошло слово «алгоритм?». Давайте послушаем сообщение (к доске выходит студент и читает сообщение)

Приложение 1.

## Сообщение: История возникновения слова «алгоритм».

Правила выполнения арифметических действий над целыми числами и дробями в десятичной системе счисления впервые были сформулированы выдающимся средневековым ученым по имени Мухаммед ибн Мусса аль – Хорезми (в переводе с арабского это означает «Мухаммед, сын Муссы из Хорезма»), сокращенно Ал – Хорезми.

Ал – Хорезми жил и творил в IX веке. Арабский оригинал его арифметического труда утерян, но имеется латинский перевод XII века, по которому западная Европа ознакомилась с десятичной системой счисления и правилами выполнения в ней арифметических действий.

Ал – Хорезми стремился к тому, чтобы сформулированные им правила были понятны для всех грамотных людей. Достичь этого в веке, когда еще была разработана математическая символика (знаки операций, скобки, буквенные обозначения и тому подобное) было очень трудно. Но Ал-Хорезми удалось выработать в своих трудах такой стиль четкого, строгого словесного предписания, который не давал читателю никакой возможности уклониться от предписанного или пропустить какие – нибудь действия.

В латинском переводе книги Ал-Хорезми правила начинались словами «Алгоризми сказал». С течением времени люди забыли, что «Алгоризми» - это автор правил, и стали сами правила называть алгоритмами. Постепенно «Алгоризми сказал» преобразовалось в «алгоритм гласит».

Таким образом, слово «алгоритм» происходит от имени ученого Ал-Хорезми. Как научный термин первоначально оно обозначало лишь правила выполнения действий в десятичной системе счисления. С течением времени это слово приобрело более широкий смысл и стало обозначать любые точные правила действий.

Научное определение понятия алгоритма дал А. Черч в 1930 году.

В настоящее время слово «алгоритм» является одним из важнейших понятий науки информатики и имеет следующее определение:

Алгоритм – это описание последовательности действий (план), строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов.

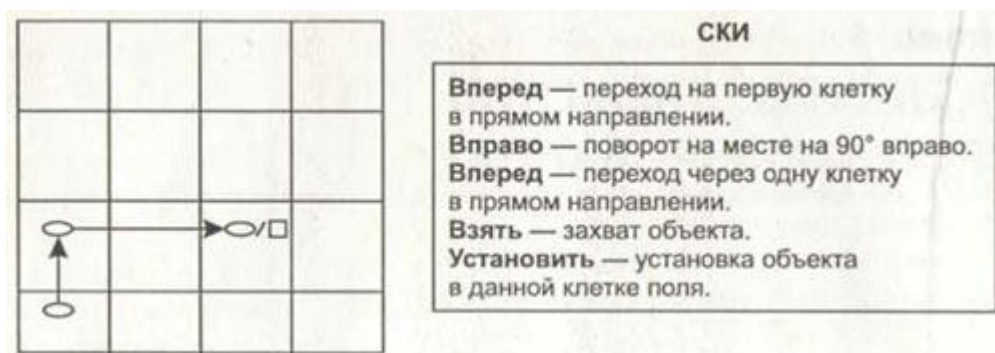
С понятием «алгоритм» тесно связано понятие «исполнитель». Чтобы достичь цели, алгоритм должен быть кем-то или чем-то исполнен. Это может быть человек, механическое устройство, робот, компьютер и другие, способные понимать и выполнять команды алгоритма. Каждый исполнитель имеет свою систему команд – конечный перечень доступных пониманию указаний. Эта совокупность команд называется системой команд исполнителя (СКИ).

Пример. Робот должен выполнить следующее задание: продвигаясь по цеху, взять деталь и установить ее.

Представим цех в виде клеток (рабочих полей), по которым должен продвигаться робот. Стрелки указывают направление его движения (рис.1).

Исполнитель действует формально (не вникая в содержание, выполняет некоторые правила, инструкции), но получает требуемый результат. Следовательно, алгоритм формализует процесс решения задачи, позволяя механически использовать команды алгоритма в указанной последовательности. Итак, алгоритм – понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленной цели.

Рис. 1



Рассмотрим информационный процесс редактирования текста. При работе с текстом возможны

различные операции: удаление, копирование, перемещение или замена его фрагментов. Что необходимо для того, чтобы преобразовать текст?

- Первое. Требуется исполнитель.
- Второе. Процесс должен быть разбит на этапы, понятные исполнителю.
- Третье. Должно быть определено начальное состояние текста и его требуемое конечное состояние.

Теория алгоритмов имеет большое практическое значение. Алгоритмический тип деятельности важен не только как одна из эффективных форм труда человека. Через алгоритмизацию, через расчленение сложных действий на все более простые, на действия, выполнение которых доступно машинам, пролегает путь к автоматизации различных процессов.

Алгоритм обладает следующими свойствами:

1. Понятность – исполнитель алгоритма должен знать, как его выполнять.
2. Конечность – выполняемый алгоритм должен приводиться к результату за конечное число шагов.
3. Дискретность – любой алгоритм должен состоять из конкретных действий, следующих в определенном порядке.
4. Массовость – один и тот же алгоритм можно использовать с различными исходными данными.

Существуют алгоритмы работы с величинами - числовыми, символьными, логическими и алгоритмы работы «в обстановке» (например робот).

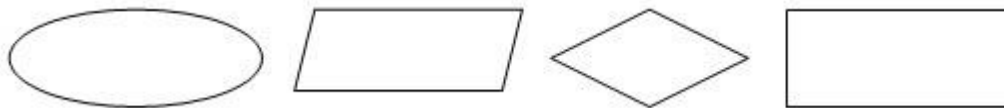
Они должны быть записаны на алгоритмическом языке, т.е. представлены в виде графического и (или) словесного описания, таблицы, последовательности формул, на учебном алгоритмическом языке, языке программирования. Иногда их записывают на псевдокодах (специальных языках).

Алгоритмический язык - это система обозначений и правил для единообразной и точной записи алгоритмов и их исполнения. Язык записи алгоритма должен быть понятным для человека.

Наиболее распространен специальный графический язык описания алгоритмов - структурные схемы. Схема в стандарте определена как «графическое представление определения, анализа или метода решения задачи, в котором используются символы для отображения операций, данных, потока, оборудования и т.д.» Существуют схемы данных, программ, работы системы, взаимодействия программ, ресурсов системы. Схемы обеспечивают наглядность, читаемость, отображение последовательности выполняемых процессов.

Схема - это ориентированный граф, стрелками (или линиями) указывающий порядок исполнения команд алгоритма, а вершины (события) такого графа представлены геометрическими фигурами, которые называются символами. Например, начало и конец записи алгоритма обозначаются овалом, данные (носитель которых не определен) - параллелограммом, процесс (обработка данных любого вида) - в виде прямоугольника, решение (или функция переключа-тельного типа, например условие) - в виде ромба и т. д. (рис. 2). Стандарт на этот специальный графический язык для записи дан в ЕСПД (Единая система программной документации, стандарт - ГОСТ 19.701-90).

Рис. 2



В основе построения алгоритмических структур лежит теорема структурного программирования, включающая следующие основные принципы:

- всякий реальный алгоритм может быть построен с использованием трех базовых элементов: следования, ветвления и цикла;
- любая алгоритмическая структура, состоящая из базовых элементов, может быть представлена как единый процесс;
- алгоритм проектируется по нисходящей схеме;
- поэтапное уточнение или пошаговая детализация алгоритма.

В графической форме базовые вершины могут быть одного из трех типов:

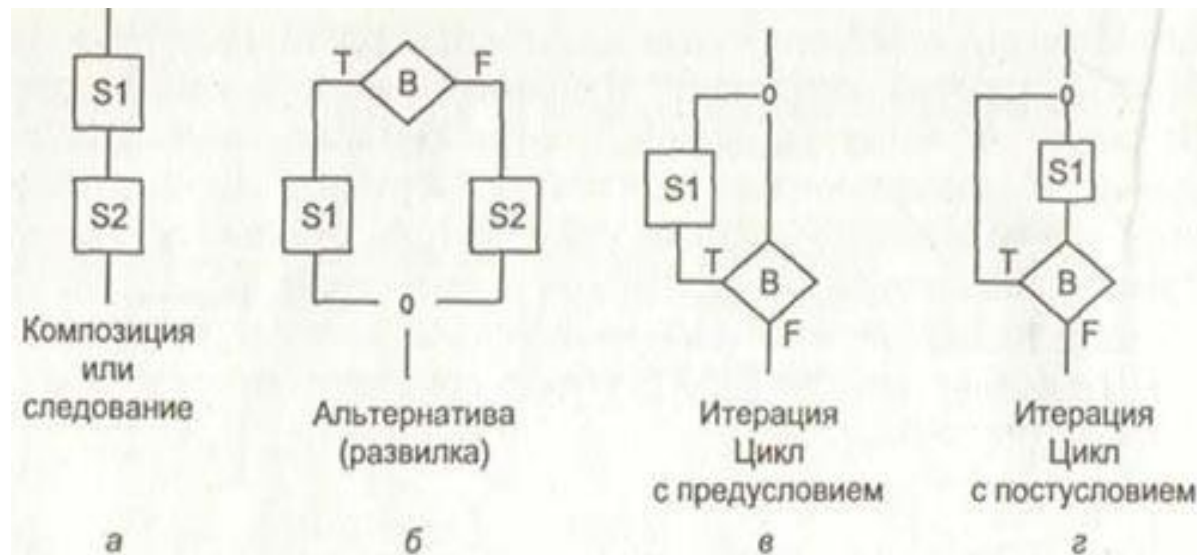
- функциональная (один вход и один выход);

- предикатная (один вход и два выхода);
- объединяющая (два выхода и один вход, передающий управление от первого из двух выходов).

Из данных элементарных схем можно построить четыре схемы основных алгоритмических структур, имеющих особое значение для практики алгоритмизации.

Схема а на рис. 3 - самая простая структура алгоритма - это последовательность, если команды следуют одна за другой в естественном порядке. Такой алгоритм называется линейным. Здесь S1 и S2 - некоторые серии команд (S - функциональные вершины).

Рис. 3. Схемы основных алгоритмических структур:



а - линейный алгоритм; б – разветвляющий алгоритм;

в,г – циклические алгоритмы с предусловием и постусловием соответственно

Схема б на рис. 3 - структура, в которой порядок следования команд определяется в зависимости от результатов проверки некоторых условий. Такой алгоритм называется разветвляющимся. В алгоритме разветвляющейся структуры действия записываются не подряд, а в зависимости от выполнения (невыполнения) некоторого условия. Здесь В - условие (предикатная вершина), в зависимости от истинности (Т) или ложности (F) которого управление передается по одной из двух ветвей. Объединяющая вершина обозначена - О.

Схемы в, г на рис. 3 - структуры, в которых получение результата обеспечивается многократным повторением одних и тех же операций. Это - циклические алгоритмы с предусловием и постусловием.

Схему (блок-схему) алгоритма можно изобразить, например, вызвав через меню текстового процессора MSWord команды: Вид " Панели инструментов " Рисование с помощью Автофигур (группа Блок-схема), где представлены начертания всех стандартных фигур и их обозначения. Надписи в фигурах следует создать с помощью команд: Вставка "Надпись".

Типовая блок-схема алгоритма линейной структуры показана на рис. 4. Типовая блок-схема алгоритма разветвляющейся структуры представлена на рис. 5.





Рис. 4



Рис. 5

В схеме разветвления алгоритма операцию проверки условия выполняет логический блок, изображенный ромбом, внутри которого указывается проверяемое условие (отношение), и имеется два выхода: ДА и НЕТ. Если условие истинно, то выходим на ДА, если ложно - то НЕТ. Типовая блок-схема алгоритма циклической структуры показана на рис. 6.

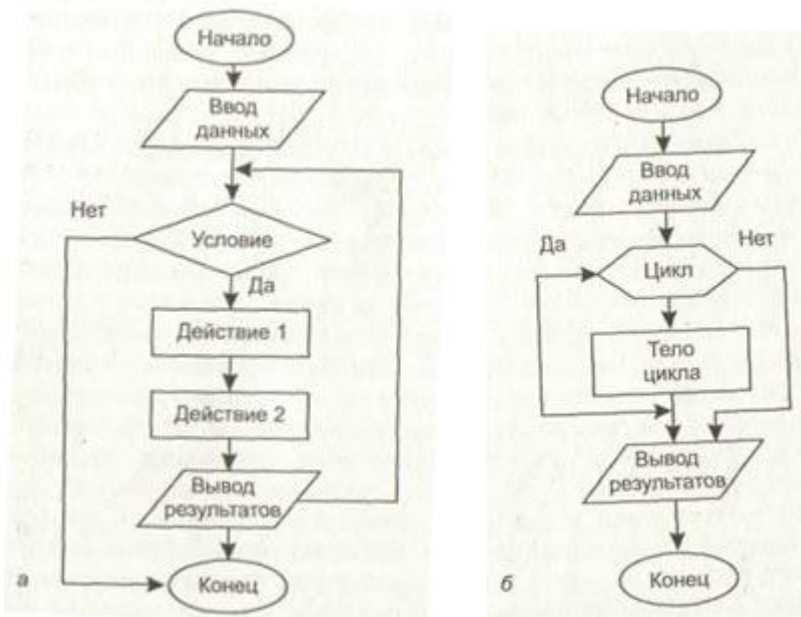


Рис. 6. Блок-схема алгоритмов циклической структуры:

а - цикл с предусловием; б - цикл с постусловием

Для составления алгоритма следует:

1. осмыслить условия (требования) задачи, т.е. выяснить, что является исходными данными, какие из них допускаются в задаче, что является результатом;
2. составить строгую формулировку задачи по следующей форме: задача, аргументы, ограничения, результаты;
3. записать описательную часть алгоритма и наметить план решения, т.е. процедурную часть, составив ее в произвольной словесной форме;
4. переписать алгоритм, используя допустимые для предполагаемого исполнителя команды, оставляя неясные места;
5. многократно переписывать алгоритм, каждый раз уделяя внимание вопросам, оставшимся от предыдущего шага, более мелким деталям, пока не получится текст, по которому можно писать программу для исполнителя (не обязательно компьютера).

Чтобы алгоритм стал понятен ЭВМ, его следует закодировать - перевести на строго

формализованный язык, т.е. язык программирования. Кодирование - процесс достаточно механический, но требует твердого знания команд и синтаксиса языка программирования.

Алгоритм, записанный на языке программирования, называется программой для ЭВМ. Языки эти формальные, специально созданные для общения человека с компьютером. Каждый язык программирования имеет алфавит, словарный запас, грамматику, синтаксис и семантику.

Алфавит - фиксированный набор символов, допускаемых для составления текста программы на этом языке.

Синтаксис - система правил, определяющих допустимые конструкции языка программирования из букв алфавита.

Семантика - система правил однозначного толкования отдельных языковых конструкций, позволяющих воспроизвести выполнение процесса обработки данных. Если программа написана в машинных кодах, то она может сразу исполняться ЭВМ. Но писать такие программы сложно, они очень громоздки и плохо воспринимаются человеком. Если формальный язык, на котором написана программа, хорошо понятен человеку, но не может сразу быть воспринят компьютером, то создают специальные программы (трансляторы: компиляторы или интерпретаторы), осуществляющие автоматический перевод программ с языка программирования в машинные коды.

Предположим, что исполнителем алгоритма будет ЭВМ, т. е. электронный автомат. Компьютер, к сожалению, не воспринимает команды на естественном языке. Для управления компьютером необходимо освоить специальный командный язык. Внутри себя компьютер составляющими его микроэлектронными устройствами командует на внутреннем машинном языке, удобном для микроэлектроники. Людям трудно общаться на этом языке, его понимают только специалисты. Для общения между компьютером и человеком, точнее, для приказов компьютеру что-то делать, изобретены специальные языки, названные языками программирования. На этих языках составляются программы - совокупности команд для управления компьютером.

Программа - упорядоченный список команд (инструкций, включающих операторы и параметры) на языке программирования. Такая программа называется исходным текстом или исходным кодом. Для реализации программы она должна быть откомпилирована, в результате чего образуется объектный код, записанный в машинных кодах. Для подключения к программе необходимых стандартных процедур и функций используют программу редактор связей, которая выполняет эту работу, извлекая из библиотек стандартных подпрограмм необходимые и вставляя их в объектный код. Полученная программа называется исполняемым кодом и является уже рабочей программой, которую можно запустить на исполнение. Для записи исходного текста программы прежде всего необходимы три простые команды: присваивание, ввод, вывод. Команда присваивания служит для изменения состояния объектов алгоритма и обозначается, на пример в языке Pascal, символом := (например, X:=1, A:=B). Команды ввода-вывода необходимы для связи ЭВМ с внешним миром. Эти команды относятся к разряду основных и реализованных в любой машине. Основной единицей программы, выполняющей определенные действия над данными, является оператор. Операторы бывают управляющие и обрабатывающие.

В программах для ЭВМ используются описания объектов следующих типов: целые, вещественные, литерные (символьные) и др. С каждым типом связан свой способ представления объекта в памяти ЭВМ.

Основная структура программы на языке Паскаль имеет следующие разделы: заголовок, описание данных, начало программы, операторы, конец программы.

Здесь приведен пример записи алгоритма на учебном алгоритмическом языке и программы на языке Паскаль для решения конкретной задачи.

Пример. Найти периметр и площадь круга радиуса R.

Аргументы:

R - радиус круга, вещественное число.

Результаты:

L-периметр круга, вещественное число;

S-площадь круга, вещественное число.

Алг Парам (вещR, L, S)

Арг R

Рез L, S

Нач

Ввод (R)

Определить периметр  $L := 2\pi R$

Определить площадь  $S := \pi R^2$

Вывод(L,S)

Кон

Program Prim-1 (Input, Output);

const

Pi=3.1416

var

R, L, S: real;

begin

readln (R);        {ввод R}

L:=2\* Pi\*R;        {вычислениеL}

S:=Pi\*sqr (R);     {вычисление S}

writeln (L,S);     {выводLiS}

end.

Программы, записанные на любом языке программирования, сначала с помощью трансляторов переводят в машинный код. А с помощью программ-отладчиков, позволяющих находить ошибки в программе, можно посмотреть во время работы ее машинные коды. Машинный код записывается в шестнадцатеричной системе счисления, двузначными числами или обозначениями (например, 33, 41, 45, CO, F6 и т. д.), каждому из которых отведен байт, находящийся в своей ячейке памяти. Таким образом, программа в машинном коде - это набор байтов, которые процессор понимает и их различает: команды, числа, символы, адреса.

Более понятным программистам, чем машинный код, является специальный код - код ассемблера. Он записывается в виде мнемоник. Каждая команда - мнемоника (сокращенные слова английского языка). Например, машинный код, представленный числом 93 в виде мнемоники, записывается как EXCHG BX, AX и означает: обменять (EXCHG) содержимое регистров BX и AX. Иногда системные программы пишут на ассемблере, а потом переводят в машинный код - ассемблирование или наоборот, чтобы легче было читать программу, выполняют дисассемблирование.

Постепенно программы накапливались, особенно на процессы, выполняемые наиболее часто (например ввод данных с клавиатуры и вывод информации на экран и т. д.). Такие небольшие программы (процедуры) называют стандартными. В настоящее время программист при создании большой программы обращается к библиотекам стандартных программ, извлекает необходимые программы-процедуры и включает их в свою программу. Это значительно облегчает процесс программирования.

С самого начала следует выработать хороший стиль программирования и оформления программ: при создании давать им содержательные имена (идентификаторы), использовать ступенчатую форму записи, каждый оператор размещать на отдельной строке, давать ясные комментарии, размещать текст программы только в рамках экрана и т.п.

Закрепление темы.

Мы с вами познакомились с видами и свойствами алгоритмов. Теперь я предлагаю вам закрепить полученные на сегодняшнем занятии знания и выполнить самостоятельную работу. Вам необходимо будет рассмотреть задачу, определить вид алгоритма и записать его при помощи одной из форм записи алгоритмов (практикум по решению задач, приложение 2).

## ПРИЛОЖЕНИЕ 2

### Практикум по решению задач

**Задача 1.** Злоумышленник поменял местами действия в алгоритме вычисления среднего арифметического из квадратного корня трёх чисел:

Присвоить  $a$  значение  $(a^2 + b^2 + c^2) / 3$ .

Вести  $a, b, c$

Сообщить “Среднее арифметическое квадратов равно”

Сообщить  $a$ .

Восстановите правильный порядок действий.

**Задача 2.** Исправьте следующий алгоритм решения уравнения  $(x-2)(x+2) = 0$ :

Присвоить  $x$  значение  $+2$ .

Сообщить “Корни уравнения равны”.

Сообщить первое значение  $x$ .

Сообщить второе значение  $x$ .

**Задача 3.** Автомобиль проехал три участка пути разной длины с разными скоростями. Составьте алгоритм нахождения средней скорости автомобиля.

**Задача 4.** Запишите в виде алгоритмов правила определения знака:

А) произведения двух действительных чисел;

Б) суммы двух действительных чисел.

6. В записи алгоритма вычисления значения выражения  $(x^2 - 5x + 5) / (x^6 - 4x^2 + 3)$

Злоумышленник одно действие поставил не на свое место. Вот как стал выглядеть алгоритм:

вести  $x$

если  $x^6 - 4x^2 + 3 = 0$ , то:

сообщить “При таком  $x$  значение выражения не определено”.

иначе:

присвоить  $y$  значение  $(x^2 - 5x + 5) / (x^6 - 4x^2 + 3)$ .

конец ветвления.

сообщить  $y$ .

Верните действие на свое место.

**Задача 5.** Вывести на экран 3 строки текста:

1) Сказка Репка; 2) Посадил дед Репку); 3) Выросла Репка большая-пребольшая.

**Задача 6.** Вычислить значения корней квадратного уравнения:  $x^2 + px + q = 0$

Электронный тест в тестовой оболочке Miraxtest.

Подведение итогов: выставление оценок с учетом процентного выполнения теста.

Приложение 3.

## ПРИЛОЖЕНИЕ 3

## *Электронный тест в тестовой оболочке MyTestEditor*

*1. Алгоритм — это:*

- а) правила выполнения определенных действий;*
- б) ориентированный граф, указывающий порядок исполнения некоторого набора команд;*
- в) понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленных целей;*
- г) набор команд для компьютера;*
- д) протокол вычислительной сети.*

*2. Укажите наиболее полный перечень способов записи алгоритмов:*

- а) словесный, графический, псевдокод, программный;*
- б) словесный;*
- в) графический, программный;*
- г) словесный, программный;*
- д) псевдокод.*

*3. Суть такого свойства алгоритма как результативность заключается в том, что:*

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);*
- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;*
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;*
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;*
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.*

*4. Суть такого свойства алгоритма как массовость заключается в том, что:*

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);*
- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;*
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;*
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;*
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.*

*5. Суть такого свойства алгоритма как дискретность заключается в том, что:*

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);*
- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;*
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;*
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;*
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.*

*6. Суть такого свойства алгоритма как понятность заключается в том, что:*

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);*

- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.

7. Суть такого свойства алгоритма как детерминированность заключается в том, что:

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
- б) записывая алгоритм для конкретного исполнителя можно использовать лишь те команды, что входят в систему его команд;
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.

8. Алгоритм называется линейным:

- а) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- б) если ход его выполнения зависит от истинности тех или иных условий;
- в) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
- г) если он представим в табличной форме;
- д) если он включает в себя вспомогательный алгоритм.

9. Алгоритм называется циклическим:

- а) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- б) если ход его выполнения зависит от истинности тех или иных условий;
- в) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
- г) если он представим в табличной форме;
- д) если он включает в себя вспомогательный алгоритм.

10. Алгоритм включает в себя ветвление, если:

- а) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- б) если ход его выполнения зависит от истинности тех или иных условий;
- в) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
- г) если он представим в табличной форме;
- д) если он включает в себя вспомогательный алгоритм.

11. Системы программирования:

- а) обеспечивают непосредственное решение пользовательских задач;
- б) позволяют создавать новые программы на языках программирования;
- в) обеспечивают работу всех аппаратных устройств компьютера и доступ пользователя к ним;
- г) представляют собой совокупность программ, используемых для различных операций с документами;
- д) предназначены для уничтожения компьютерных вирусов и зараженных ими файлов.

12. Характерным признаком линейной программы является:

- а) строго последовательное выполнение операторов в порядке их записи;
- б) наличие в каждой программной строке только одного оператора;
- в) использование в ней исключительно операторов присваивания;
- г) присутствие в ней операторов условного и безусловного перехода;
- д) присутствие в ней операторов цикла.

13. Подпрограммой называют:

- а) часть программы, содержащую неоднократно выполняемые команды;
- б) независимый программный модуль;
- в) произвольный фрагмент программы;
- г) набор операторов, следующий в программе за оператором GOSUB;
- д) часть программы, служащей для решения некоторой вспомогательной задачи.

14. Важнейший принцип структурного программирования базируется на утверждении:

- а) любой алгоритм имеет дискретную структуру;
- б) алгоритм любой сложности можно построить с помощью следующих базовых структур: линейной, ветвящейся, циклической;
- в) современный компьютер — это единство аппаратных средств и программного обеспечения;
- г) сущность формализации решаемой задачи заключается в составлении алгоритма;
- д) в качестве обязательного этапа создания программы выступает ее тестирование и отладка.

15. Переменная в программировании полностью характеризуется:

- а) именем;
- б) именем, значением и типом;
- в) именем и типом;
- г) именем и значением;
- д) значением.

Задание на дом: выучить определения, привести примеры алгоритмов из жизненной практики.