

Последовательность
(Жуйелілік, реттілік)
(Sequence)

Последовательность

- совокупность идущих **один из другим** объектов или действий;
- **упорядоченная** совокупность элементов;
- числа или элементы, расположенные **в организованном порядке**;
- постоянство, преемственность, логичность, ряд, прогрессия.

Приготовить горячий чай

Приготовить горячий чай

1. Взять посуду
2. Наполнить водой
3. Поставить кипятиться
4. Взять заварку (чай)
5. Взять чайник
6. Заварить чай (заварка+кипяченая вода)

Методика изучения основ алгоритмизации и программирования

Лектор: **Ергалиев Ерлан**

Перечень понятий

- *Алгоритм*
- *Свойства алгоритмов*
- *Исполнители алгоритмов*
- *Система команд исполнителя*
- *Программирование*
- *Языки программирования*
- *Вспомогательные алгоритмы*

Цель изучения темы

Изучение алгоритмизации в школьной информатике может иметь два целевых аспекта:

- ✓ развивающий аспект, под которым понимается развитие алгоритмического мышления учащихся;
- ✓ программистский аспект.

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

АЛГОРИТМИЗАЦИЯ

АЛГОРИТМ — последовательность команд управления работой исполнителя

Исполнитель алгоритма

Характеристики исполнителя: назначение, среда, режимы работы, система команд — СКИ

Свойства алгоритма: понятность, точность, конечность, дискретность, массовость

Алгоритмы работы «в обстановке»

Алгоритмы работы с величинами

Величина: имя, значение, тип

Базовые алгоритмические структуры

Следование

Ветвление

Цикл

Методика структурного программирования

— суперпозиция базовых структур (последовательность, вложенность)
— нисходящее проектирование сложных алгоритмов

Языки описания алгоритмов: блок-схемы, учебный алгоритмический язык

ПРОГРАММИРОВАНИЕ

Функциональное (Лисп)

Логическое (Пролог)

Объектно-ориентированное (Смолток, СИ++, Делфи ...)

Процедурное программирование (Ассемблеры, Бейсик, Фортран, Паскаль...)

Организация ввода, редактирования, сохранения

Организация трансляции и компоновки программных модулей

Организация отладки и исполнения программ

Языки программирования

Элементы языка

Организация действий над данными

Организация данных

Алфавит, лексемы, синтаксис, оформление

Ввод/вывод данных, работа с файлами

Обработка данных

Типы и структуры данных

Операции и выражения

Операторы: присваивания, ветвления, циклов, переходов ...

Организация и использование подпрограмм

Алгоритм

- Алгоритмом называется **точное и понятное предписание** исполнителю совершать последовательность действий, направленных на решение поставленной задачи.
- Алгоритм – это **описание способа решения задачи**, в котором предусматривается разбиения процесса решения на конечную по времени последовательность действий, представленных в виде элементарных операций.

Алгоритм

- **набор инструкций**, описывающих порядок действий исполнителя для достижения некоторого результата;
- **совокупность последовательных шагов**, схема действий, приводящих к желаемому результату;
- **система последовательных операций** (в соответствии с определёнными правилами) для решения какой-нибудь задачи.

Исполнитель

- **Субъект**, выполняющий алгоритм.
- **Объект (или субъект)**, для управления которым составляется алгоритм.
- Основной характеристикой исполнителя, с точки зрения управления, является *система команд исполнителя* (СКИ). Это конечное множество команд, которые понимает исполнитель, т.е. умеет их выполнять.
- Робот, компьютер и человек.

Схема функционирования исполнителя



Свойства алгоритмов

Алгоритм обладает следующими основными свойствами:

- дискретностью;
- определенностью (детерминированностью, точностью);
- массовостью;
- результативностью;
- формальностью.

Дискретность

Свойство алгоритма, которое характеризует его структуру. Любой алгоритм состоит из отдельных операций (этапов, действий), которые выполняются дискретно (по шагам). Это означает, что алгоритм обладает свойством дискретности.

Детерминированность

Свойство алгоритма, указывающее на то, что каждый шаг алгоритма должен быть строго определен и не может допускать различных толкований. Также строго должен быть определен порядок выполнения отдельных шагов, то есть исполнитель должен точно знать последовательность выполнения операций. Любой алгоритм должен быть представлен таким образом, чтобы он мог быть однозначно (точно) реализован исполнителем. Это свойство алгоритма называют также определенностью, однозначностью или точностью.

Массовость (универсальность)

Применимость алгоритма ко всем задачам рассматриваемого типа при любых допустимых множествах исходных данных. Здесь важно подчеркнуть, что массовость означает применимость алгоритма ко всем задачам *рассматриваемого* типа, то есть ко всем задачам, для решения которых он предназначен. Кроме того, здесь необходимо иметь в виду, что реализация алгоритма возможна при любых, но *допустимых* множествах исходных данных.

Результативность (конечность)

Способность получения определенного результата для допустимых исходных данных за конечное число шагов. То есть способность завершать процесс за конечное число итераций или формировать сообщение о невозможности дальнейшей обработки данных (например, в связи с тем, что к имеющимся исходным данным этот алгоритм не применим).

Формальность

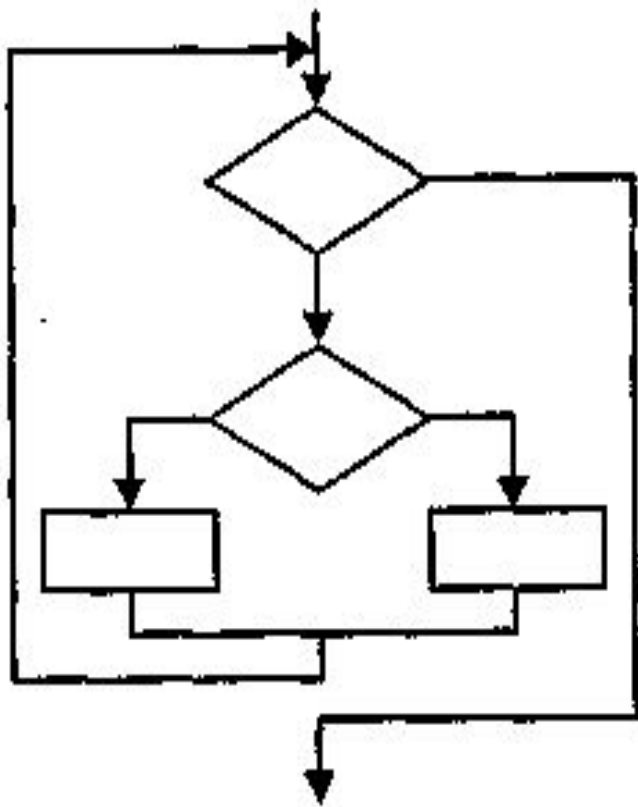
Свойство означающее, что любой исполнитель, выполняющий алгоритм (например, компьютер), действует формально, то есть строго выполняет инструкции предусмотренные разработчиком алгоритма.

Способы описания алгоритмов

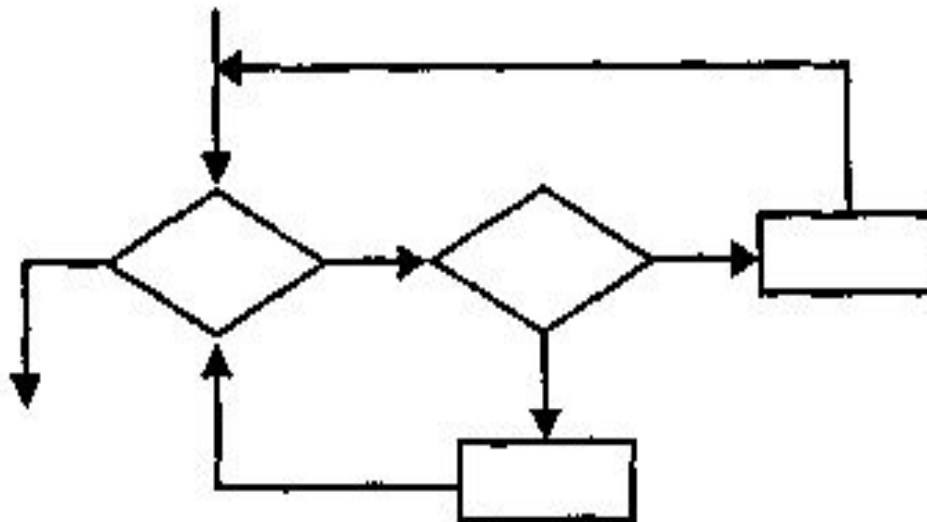
Блок-схемы – схема, описывающая алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности.

Учебный алгоритмический язык - это текстовая форма описания алгоритма.

Блок-схема



a



б

Приготовить горячий чай

1. Взять посуду
2. Наполнить водой
3. Поставить кипятиться
4. Взять заварку (чай)
5. Взять чайник
6. Заварить чай (заварка+кипяченая вода)

Составьте алгоритм
действий для выполнения
домашней работы

Выполнение домашней работы

1. Определение задачи
2. Определение условий выполнения
3. Ознакомление с заданием
4. Разделения задания на части
5. Начало выполнения первой части задания
6. Процесс выполнения
7. Завершение
8. Проверка результатов на соответствия условий выполнения
9. Заключение исходя из результатов задачи

Программирование

Процесс подготовки задач для решения их на ЭВМ, состоящий из следующих этапов:

- **составление** "плана решения" задачи в виде набора операций (алгоритмическое описание задачи);
- **описание** "плана решения" на языке программирования (составление программы);
- **трансляция** программы с языка программирования на машинный язык (в виде последовательности команд, реализация которых техническими средствами ЭВМ и есть процесс решения задачи).

Программирование

- Это раздел информатики, задача которого — **разработка** программного обеспечения ЭВМ.
- **процесс разработки** программы на определенном языке программирования.

Парадигмы программирования

- процедурное программирование (Паскаль, Бейсик, Фортран, Си, Ассемблеры)
- логическое программирование (Пролог)
- функциональное программирование (Лисп)
- объектно-ориентированное программирование (Смолток, Си++, Делфи).

Языки программирования



Низкий уровень

Среди характеристик часто встречаются:

- ✓ ограничения на абстракции данных
- ✓ Сильная статическая типизация
- ✓ отсутствие промежуточной среды выполнения
- ✓ прямой доступ к памяти.



Средний уровень

Среди характеристик часто встречаются:

- ✓ фокус на абстракциях
- ✓ сильная статическая типизация
- ✓ среда выполнения
- ✓ ограничения на прямой доступ к памяти.



Высокий уровень

Среди характеристик часто встречаются:

- ✓ сильное абстрагирование
- ✓ динамическая и/или слабая типизация
- ✓ полностью независимое управление памятью и/или наличие среды выполнения.



Процесс изучения и практического освоения программирования можно разделить на три части

- изучение методов построения вычислительных алгоритмов
- изучение языка программирования
- изучение и практическое освоение определенной системы программирования.

Требования к уровню подготовки

Учащиеся должны:

- понимать сущность понимания алгоритма, знать его основные свойства, иллюстрировать их на конкретных примерах алгоритмов;
- понимать возможность автоматизации деятельности человека при исполнении алгоритмов;
- знать основные алгоритмические конструкции и уметь использовать их для построения алгоритмов;
- определять возможность применения исполнителя для решения конкретной задачи по системе его команд, построить и исполнить на компьютере алгоритм для учебного исполнителя (типа «черепахи», «робота» и т.д.);
- записать на учебном алгоритмическом языке (или языке программирования) алгоритм решению простой задачи.

Обязательный минимум содержания учебного материала

Изучение учебного материала данной содержательной линии курса обеспечивает учащимся возможность:

- понять (на основе анализа примеров) смысл понятия алгоритма, знать свойства алгоритмов, понять возможность автоматизации деятельности человека при исполнении алгоритмов;
- освоить основные алгоритмические конструкции (цикл, ветвление, процедура), применять алгоритмические конструкции для построения алгоритмов решения учебных задач;
- получить представление о «библиотеке алгоритмов», уметь использовать библиотеку для построения более сложных алгоритмов;
- получить представление об одном из языков программирования (или учебном алгоритмическом языке), использовать этот язык для записи алгоритмов решения простых задач.

6 класс

Алгоритмизация и моделирование (8 часов):

- ✓ алгоритмы и их исполнители
- ✓ формы записи алгоритмов
- ✓ блок-схемы
- ✓ типы алгоритмов (линейные, разветвляющиеся и циклические алгоритмы)
- ✓ модель, как отражение существенных свойств реального объекта
- ✓ виды моделей.

6 класс

ГЛАВА 3. АЛГОРИТМИЗАЦИЯ И МОДЕЛИРОВАНИЕ	55
§8. Алгоритмы и их исполнители	55
§9. Формы записи алгоритмов.....	58
§10. Типы алгоритмов.....	62
§11. Циклические алгоритмы.....	67
§12. Практикум по составлению алгоритмов.....	70
§13. Модель и моделирование	72
§14. Виды моделей.....	77
§15. Практикум по моделированию.....	81

Учащиеся 6 класса должны

Знать:

- понятие алгоритма и исполнителя;
- формы записи алгоритма;
- запись алгоритма блок-схемами;
- типы алгоритмов;

Уметь:

- составлять алгоритмы разного вида в словесной и графической формах;

7 класс

Алгоритмизация и программирование (9 часов):

- ✓ программирование как формальный способ записи алгоритмов
- ✓ синтаксис языка программирования
- ✓ типы переменных
- ✓ правила записи арифметических выражений
- ✓ программирование линейных алгоритмов
- ✓ оператор ввода и вывода
- ✓ оператор присваивания.

7 класс

ГЛАВА 3. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PASCAL	42
§8. Языки программирования	42
§9. Программа и ее структура	55
§10. Данные	62
§11. Организация ввода и вывода данных	68
§12. Выражения	74
§13. Числовые функции	80
§14. Программирование линейных алгоритмов	83
§15–16. Практикум по теме «Программирование линейных алгоритмов»	89

Учащиеся 7 класса должны

Знать:

- понятие языка программирования;
- структуру программы на языке программирования;
- синтаксис языка программирования;
- типы переменных;
- правила записи арифметических выражений на языке программирования;
- операторы ввода, вывода и присваивания;

Уметь:

- определять типы переменных;
- составлять программы линейной структуры;

8 класс

Алгоритмизация и программирование (10 часов):

- операторы ветвления;
- сложные условия;
- циклические алгоритмы,
- программирование циклических алгоритмов,
- циклы с прямым и обратным счетчиком,
- цикл с предусловием;
- цикл с постусловием,
- графические операторы и процедуры.

8 класс

.....	45
ГЛАВА 3. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ	53
§ 11. Программирование условий.....	53
§ 12. Сложные условия.....	65
§ 13. Практикум по теме «Программирование алгоритмов структуры ветвления».....	74
§ 14. Графический режим.....	77
§ 15. Процедуры записи текстов в графическом режиме.....	87
§ 16. Оператор while..do..	90
§ 17. Оператор repeat..until..	97
§ 18. Оператор for	102
§ 19–20. Практикум по теме «Организация циклов».....	108

Учащиеся 8 класса должны

Знать:

- операторы программирования разветвляющих алгоритмов;
- операторы условного и безусловного перехода;
- операторы программирования циклических алгоритмов;
- циклы с параметром, циклы ДО, цикл ПОКА;

Уметь:

- составлять программы на основе алгоритмов ветвления и циклов;
- использовать графический режим работы в среде программирования;

9 класс

Алгоритмизация и программирование (16 часов):

- файлы и их обработка, структурированные данные;
- вложенные циклы;
- понятие о массивах, стандартные алгоритмы обработки массивов;
- символьные строки, посимвольная обработка строк, функции для работы с символьными строками.

9 класс

ГЛАВА 1. АЛГОРИТМИЗАЦИЯ	6
§2. Понятие алгоритма.....	11
§3. Этапы решения задач.....	11
ГЛАВА 2. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PASCAL	20
§4. Языки программирования.....	29
§5. Программа и ее структура.....	29
§6. Данные.....	42
§7. Выражения.....	50
§8. Числовые функции.....	61
§9. Программирование линейных алгоритмов.....	67
§10. Практикум по теме «Программирование линейных алгоритмов».....	71
§11. Программирование условий.....	76
§12. Составные условия.....	78
§13. Организация выбора.....	92
§14. Работа с файлами.....	99
§15–16. Практикум по теме «Программирование алгоритмов структуры ветвления».....	105
§17. Графический режим.....	111
§18. Процедуры записи текстов в графическом режиме.....	114
§19. Операторы while... do... и repeat... until.....	122
§20. Оператор For.....	125
§21. Вложенные циклы.....	134
§22–23. Практикум по теме «Организация циклов».....	139
§24. Обработка символьных данных и строк.....	148
§25. Практикум по теме «Обработка символьных данных и строк».....	149
ГЛАВА 3. ПРОГРАММИРОВАНИЕ	158
§1. Программирование.....	159
§2. Программирование.....	159

9 класс

МАТЕРИАЛ ДЛЯ УГЛУБЛЕННОГО ИЗУЧЕНИЯ	183
1. Массивы.....	183
2. Обработка одномерного массива.....	187
3. Обработка двумерного массива.....	196
4. Практикум по теме «Массивы».....	202
ПРИЛОЖЕНИЯ	205
Приложение 1. Назначение специальных символов языка программирования Pascal.....	205
Приложение 2. Служебные (зарезервированные) слова.....	206
Приложение 3. Ошибки компиляции.....	209
Приложение 4. Работа с файлами.....	213
ГЛОССАРИЙ	215
Список использованной и рекомендуемой литературы.....	218

Учащиеся 9 класса должны

Знать:

- операции работы с файлами на языке программирования;

Уметь:

- организовывать работу с файлами на языке программирования;
- составлять программы с вложенными циклами;
- использовать массивы при составлении программ;
- программировать алгоритмы символьной обработки;
- создавать модели средствами языка программирования;

Задача

Имеется 9 монет, среди которых 1 монета фальшивая (легче остальных). Как с помощью двух взвешиваний на чашечных весах определить фальшивую монету.

Спасибо за внимание!