Тема: «Введение в алгоритмизацию. Линейные алгоритмы и формальные исполнители»

1. Понятие алгоритма и его свойства (Урок 11)

Слово **«алгоритм»** происходит от имени средневекового математика Мухаммада ибн Мусы аль-Хорезми, который сформулировал правила выполнения арифметических действий над числами.

Алгоритм — это точная и понятная последовательность команд исполнителю, определяющая процесс перехода от исходных данных к искомому результату за конечное число шагов.

Проще говоря, это план действий, рецепт или сценарий, где каждое действие расписано так просто и однозначно, что его можно выполнить, даже не задумываясь над общим смыслом.

Примеры алгоритмов из реальной жизни

Мы используем алгоритмы каждый день, даже не замечая этого:

1. **Рецепт приготовления чая:**
   - Шаг 1: Налить воду в чайник.
   - Шаг 2: Включить чайник и дождаться закипания.
   - Шаг 3: Положить чайный пакетик в чашку.
   - Шаг 4: Налить кипяток в чашку.
   - Шаг 5: Подождать 3–4 минуты.
   - Шаг 6: Достать пакетик (чай готов).
2. **Переход дороги по регулируемому переходу:**
   - Шаг 1: Подойти к краю тротуара у светофора.
   - Шаг 2: Посмотреть на сигнал светофора.
   - Шаг 3: Если горит красный — стоять и ждать.
   - Шаг 4: Когда загорится зеленый — убедиться, что машины остановились.
   - Шаг 5: Перейти дорогу.
3. **Сборка конструктора или мебели:**
   - Схема из коробки с деталями — это классический алгоритм: деталь А соединить с деталью Б винтом № 1, затем прикрепить планку В и т. д.

Кто такой «исполнитель»?

У любого алгоритма обязательно есть **исполнитель** — тот, кто эти шаги повторяет:

- **Человек** (повар на кухне, пешеход, ученик, решающий задачу по математике);
- **Животное** (собака, выполняющая разученные команды дрессировщика);
- **Техническое устройство** (робот-пылесос, стиральная машина, навигатор в телефоне, компьютер).

Для техники алгоритм должен быть абсолютно точным. Если человеку можно

сказать: *«Порежь овощи помельче»*, и он поймет, то роботу такая команда непонятна. Ему нужно указать: *«Сделай 10 надрезов с шагом 5 мм»*.

Главные свойства любого алгоритма

Чтобы инструкция действительно считалась алгоритмом в информатике, она должна обладать пятью признаками:

1. **Дискретность** (раздельность)

Алгоритм состоит из четких отдельных шагов. Нельзя переходить к шагу 3, пока не выполнен шаг 2.

2. **Понятность**

Каждое действие должно быть известно исполнителю. Если дать человеку рецепт на неизвестном языке, он не сможет его приготовить. Команда должна входить в систему команд конкретного исполнителя.

3. **Определенность (детерминированность)**

Каждый шаг должен трактоваться только одним способом, без сомнений и догадок.

Плохо: «Насыпь немного сахара».

Хорошо (алгоритмично): «Насыпь 2 чайные ложки сахара».

4. **Результативность (конечность)**

Алгоритм не может длиться бесконечно — за определенное число шагов он обязан привести к результату (или к сообщению об ошибке, если результат недостижим).

5. **Массовость**

Хороший алгоритм работает не для одного частного случая, а для целой группы похожих задач. Например, алгоритм сложения чисел «столбиком» одинаково верно сложит и $25 + 17$, и $489 + 512$.

Какие бывают алгоритмы по структуре?

В зависимости от порядка выполнения команд выделяют три базовые структуры:

- **Линейный (следование):** все действия идут строго по очереди, сверху вниз, без развилок (например: надеть носки → надеть ботинки → завязать шнурки).
- **С ветвлением (условие):** в зависимости от проверки условия выбирается одно из действий (*«ЕСЛИ на улице дождь, ТО взять зонт, ИНАЧЕ надеть кепку»*).
- **Циклический (повторение):** определенные шаги повторяются несколько раз (*«ПОКА в тетради есть пустые строчки — пиши словарные слова»* или *«Сделай 10 приседаний»*).

Зачем алгоритмы нужны в компьютере?

Любая компьютерная программа — будь то мобильная игра, умная колонка или браузер — это просто сложный алгоритм, переведенный на язык, понятный процессору (язык программирования). Компьютер берет исходные данные (нажатие кнопок на клавиатуре или координаты мыши), строго по алгоритму их обрабатывает и выдает готовый результат (картинку на экране или звук).

Рассмотрите таблицу и запомните:

Таблица 1. Свойства алгоритма

Свойство	**В чем заключается суть?**	**Пример нарушения свойства**
Дискретность	Алгоритм разбит на отдельные, завершенные элементарные шаги (команды). Следующая команда начинается только после завершения предыдущей.	Инструкция дана сплошным неструктурированным текстом без четких действий.
Детерминированность (определенность)	Каждая команда трактуется однозначно. При одних и тех же входных данных результат всегда будет одинаковым, без элемента случайности или сомнений.	«Насыпьте немного сахара» (сколько именно — грамм, ложка, стакан?).
Понятность	Алгоритм содержит только те команды, которые входят в систему команд конкретного исполнителя (СКИ).	Роботу-пылесосу дана голосовая команда «нарисуй пейзаж».
Результативность (конечность)	Алгоритм завершается за конечное число шагов и приводит к конкретному результату (либо выдает сообщение о невозможности решения).	Программа «зависла» и бесконечно перебирает одни и те же числа.
Массовость	По одному алгоритму решается не одна частная задача, а целый класс однотипных задач с разными входными данными.	Алгоритм нахождения периметра прямоугольника $P = 2 \cdot (a + b)$ подходит для любых сторон a и b, а не только для $a = 3, b = 5$.

2. Исполнители алгоритмов и их характеристики

Исполнитель — это человек, животное или техническое устройство (робот, станок, компьютер), способное выполнять заданный набор команд.

Характеристики исполнителя:

1. **Среда исполнителя (обстановка)** — пространство или условия, в которых функционирует исполнитель (клеточное поле, координатная плоскость, лабиринт, числовая прямая).
2. **Система команд исполнителя (СКИ)** — полный перечень команд,

которые данный исполнитель понимает и умеет выполнять.

3. **Отказы исполнителя**:
   - *«Не понимаю»* — поступила команда, отсутствующая в СКИ.
   - *«Не могу»* — команда входит в СКИ, но в текущей обстановке невыполнима (например, Робот пытается сделать шаг в стену).
4. **Режимы работы исполнителя**:
   - *Непосредственное управление* — человек отдает команду, исполнитель тут же ее выполняет.
   - *Программное управление* — исполнителю передается вся программа целиком, и он выполняет ее последовательно в автоматическом режиме.

Формальное исполнение алгоритма

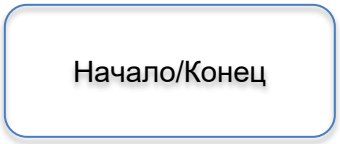
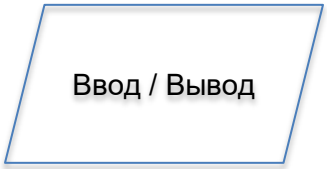
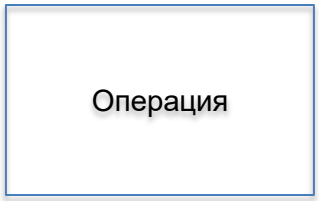
Технический исполнитель (компьютер или робот) действует **формально**: он **не задумывается над смыслом, целью или пользой** задачи, а механически выполняет команды строго по алгоритму. Если в алгоритме допущена логическая ошибка, машина все равно исполнит ее в точности так, как записано.

3. Способы записи алгоритмов и блок-схемы (Урок 11)

Существует несколько способов фиксации алгоритмов:

- **Словесный** (естественный язык: кулинарные рецепты, пошаговые инструкции).
- **Построчный / псевдокод** (алгоритмический язык с четкой структурой).
- **Язык программирования** (код на Python, Pascal, C++ и т. д.).
- **Графический (блок-схема)** — представление алгоритма в виде геометрических фигур, соединенных линиями перехода (стрелками).

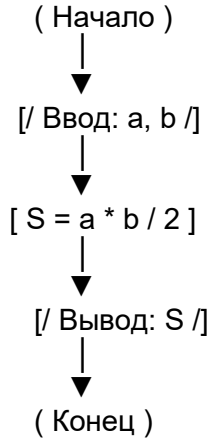
Таблица 2. Основные блоки блок-схем

Блок	Название	Назначение
	**Скругленный прямоугольник (овал)**	Обозначает старт и завершение работы алгоритма.
	**Параллелограмм**	Ввод исходных данных от пользователя или вывод итогового результата на экран.
	**Прямоугольник (процесс)**	Выполнение вычислительного действия, присваивание значений, математические расчеты.
	**Ромб (решение)**	Проверка условий (используется в ветвлениях и циклах).

4. Конструкция «Следование». Линейный алгоритм

Следование — базовая алгоритмическая конструкция, в которой команды выполняются последовательно одна за другой, в порядке их записи, без пропусков и повторений.

Линейный алгоритм — алгоритм, состоящий исключительно из конструкции «следование».



5. Практикум: формальные исполнители

В рамках практической работы № 6 мы рассмотрим виртуальные исполнители, часто встречающиеся в олимпиадах и экзаменационных материалах ОГЭ:

Исполнитель «Робот»

- **Среда:** поле, разбитое на квадратные клетки, между которыми могут стоять стены.
- **СКИ:** вверх, вниз, влево, вправо, закрасить.
- **Отказ:** разрушается, если делает шаг в стену.

Исполнитель «Чертёжник»

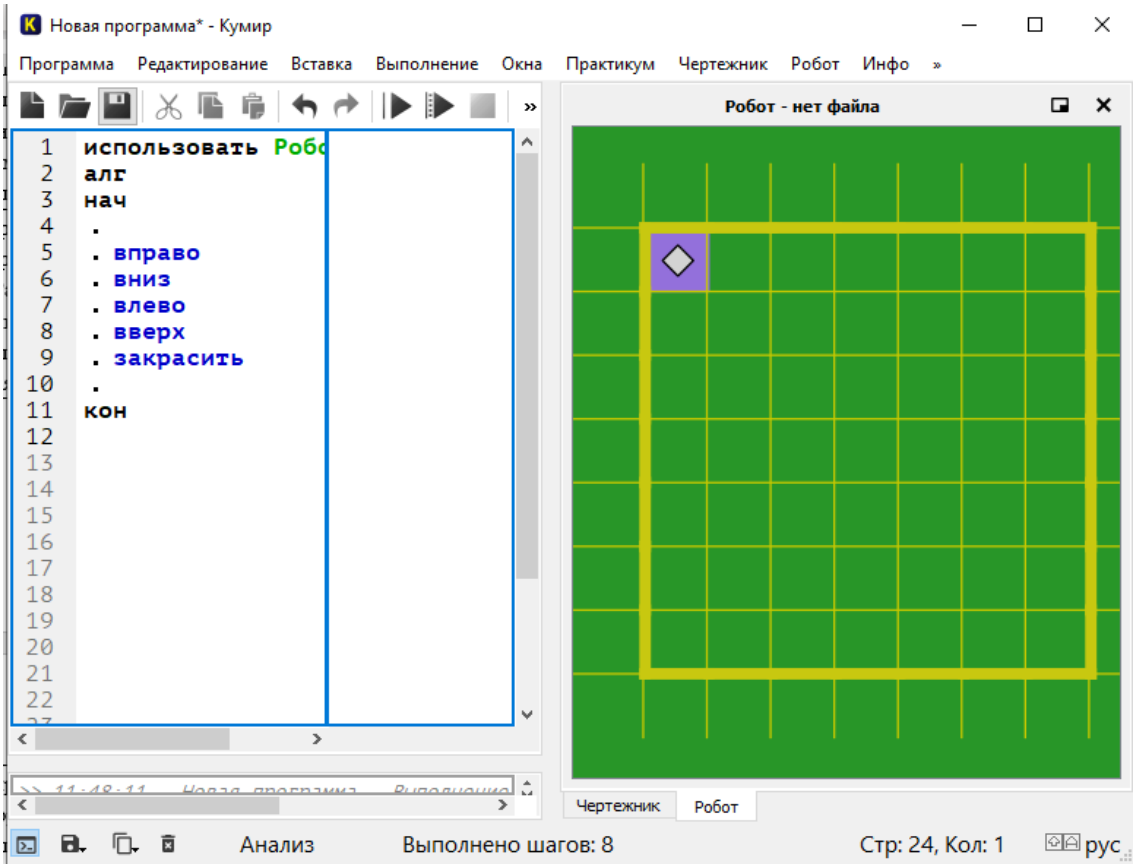
- **Среда:** декартова координатная плоскость с сеткой. Имеет перо (поднято/опущено).
- **СКИ:**
  - поднять перо, опустить перо;
  - сместиться в точку (x, y) — перемещение в абсолютные координаты (x, y);
  - сместиться на вектор (dx, dy) — перемещение относительно текущей точки на расстояние dx по горизонтали и dy по вертикали.

Исполнитель «Вычислитель»

- **Среда:** числовой регистр / числовая прямая.
- **СКИ:** набор математических преобразований исходного числа (например: 1. Прибавь 3, 2. Умножь на 2).

Задание:

1. *Установите программу с официального сайта по ссылке: https://www.niisi.ru/kumir/dl.htm*
2. *Познакомьтесь с исполнителями Робот, Чертежник*
3. *Познакомьтесь с системой команд исполнителя Робот. Для этого зайдите в меню Вставка и попробуйте вставить в программу команды: вверх, вправо, вниз, влево, закрасить. Перед началом работы необходимо вставить команду* `использовать Робот.`
4. *При нажатии на кнопку Выполнить*  *Исполнитель опишет квадрат и закрасит начальную клетку*



6. Задачи для закрепления материала

Задача 1. «Криминалист в алгоритмах»

Определите, какое фундаментальное свойство алгоритма нарушено в каждом из пунктов:

1. В инструкции по замене колеса автомобиля написано: *«Ослабьте болты, поднимите машину домкратом, покрутите колесо так, чтобы стало хорошо, снимите его»*.
2. Компьютеру с объемом свободной памяти 512 КБ отправили на печать команду: *«Загрузить файл энциклопедии размером 40 ГБ»*.
3. Программа для робота содержит команду *«Прыгни через забор»*, однако датчики и механизмы робота рассчитаны только на колесное движение.
4. В кулинарном блоге опубликован алгоритм с шагом: *«Варите крем до тех пор, пока количество пузырьков на поверхности не покажется вам идеальным»*.

1. **Детерминированность (определенность):** формулировка «чтобы стало хорошо» субъективна и не имеет точного числового критерия.
2. **Результативность (или выполнимость):** нехватка ресурсов приведет к аварийной остановке или зависанию, задача не придет к результату.
3. **Понятность:** команда отсутствует в системе команд данного исполнителя (СКИ).
4. **Детерминированность:** у исполнителя нет однозначного правила

измерения «идеальности».

Задачи на исполнителя «Вычислитель» (в формате ОГЭ)

Задача 2. «Прямой ход вычислителя»

У исполнителя «Калькулятор-2» две команды, которым присвоены номера:

1. **Прибавь 4**
2. **Умножь на 3**

Какое число получится в результате выполнения программы 11212 из исходного числа **2**?

Исходное число: 2.

1. Команда 1 (Прибавь 4): $2 + 4 = 6$
2. Команда 1 (Прибавь 4): $6 + 4 = 10$
3. Команда 2 (Умножь на 3): $10 \cdot 3 = 30$
4. Команда 1 (Прибавь 4): $30 + 4 = 34$
5. Команда 2 (Умножь на 3): $34 \cdot 3 = 102$

Ответ: 102.

Задача 3. «Обратный ход: синтез алгоритма»

У исполнителя «Альфа» есть две команды:

1. **Прибавь 2**
2. **Умножь на b**

Известно, что программа 11211 переводит число **3** в число **64**. Найдите неизвестное натуральное число b ($b \geq 2$).

Запишем шаги в виде алгебраического выражения:

1. Исходное число: 3.
2. Команда 1: $3 + 2 = 5$
3. Команда 1: $5 + 2 = 7$
4. Команда 2: $7 \cdot b = 7b$
5. Команда 1: $7b + 2$
6. Команда 1: $(7b + 2) + 2 = 7b + 4$

Составляем уравнение по условию задачи:

$$7b + 4 = 64$$

$$7b = 60$$

(Замечание: 60 *не делится нацело на* 7*. Проверим альтернативную запись команды 12111)*:

Пусть программа будет 12111 для числа 3:

1. $3 + 2 = 5$
2. $5 \cdot b = 5b$

3. $5b + 2$
4. $5b + 4$
5. $5b + 6 = 66 \Rightarrow b = 12.$

Для исходной цепочки 11211: чтобы получить целое b, изменим итоговое число на **67**:

7b + 4 = 67 => 7b = 63 => b = 9.

Ответ: 9.

Задачи на исполнителя «Чертёжник»

Задача 4. «Возвращение на базу»

Чертёжник находился в начале координат (точка $(0,0)$). Ему была отдана линейная программа:

опустить перо
сместиться на вектор (4, 6)
сместиться на вектор (-2, 3)
сместиться на вектор (5, -7)
сместиться на вектор (-3, -4)

1. В какой точке координатной плоскости окажется перо Чертёжника после выполнения алгоритма?
2. Какую **одну** команду нужно дать Чертёжнику, чтобы вернуть его перо обратно в точку $(0,0)$?

Посчитаем итоговое смещение координат как сумму компонент векторов:

$$X = 0 + 4 + (-2) + 5 + (-3) = 4$$

$$Y = 0 + 6 + 3 + (-7) + (-4) = -2$$

Перо находится в точке $(\mathbf{4}, \mathbf{-2})$.

1. Чтобы из точки $(4, -2)$ переместиться в точку $(0,0)$, требуется сдвиг на вектор $(0 - 4; 0 - (-2)) = (-4; 2)$. Команда: **сместиться на вектор (-4, 2)**.

Логическая задача на линейный алгоритм

Задача 5. «Переливания без ветвлений»

У вас есть два пустых сосуда емкостью **3 литра** и **5 литров**, а также водопроводный кран с неограниченным запасом воды и раковина для слива.

Составьте линейный алгоритм (последовательность действий), позволяющий получить в пятилитровом сосуде ровно **4 литра** воды.

Допустимые команды:

- «Наполни сосуд X до краев»
- «Опустоши сосуд X в раковину»
- «Перелей из сосуда X в сосуд Y до заполнения Y (или пока X не опустеет)»

Линейная последовательность шагов:

1. Наполни сосуд 5 л до краев. *(В сосудах: 5 л и 0 л)*
2. Перелей из сосуда 5 л в сосуд 3 л до заполнения 3 л. *(В сосудах: 2 л и 3 л)*
3. Опустоши сосуд 3 л в раковину. *(В сосудах: 2 л и 0 л)*
4. Перелей все содержимое из сосуда 5 л в сосуд 3 л. *(В сосудах: 0 л и 2 л)*
5. Наполни сосуд 5 л до краев. *(В сосудах: 5 л и 2 л)*
6. Перелей из сосуда 5 л в сосуд 3 л до его заполнения (туда поместится ровно 1 литр).

Результат: в пятилитровом сосуде осталось ровно $5 - 1 = $ ***4 литра*** *воды.*

7. Вопросы для подготовки к контрольной работе № 1

1. Может ли алгоритм состоять всего из одной команды? *(Да, если эта команда решает поставленную задачу).*
2. Чем формальный исполнитель отличается от неформального?
3. В чем разница между блоком-параллелограммом и блоком-прямоугольником в графической блок-схеме?
4. Почему линейный алгоритм называется «линейным»? Встречаются ли в нем развилки или возвраты назад?
5. Если в алгоритме поменять две любые последовательные команды местами, изменится ли результат работы? Приведите пример, когда результат изменится, и когда останется прежним.