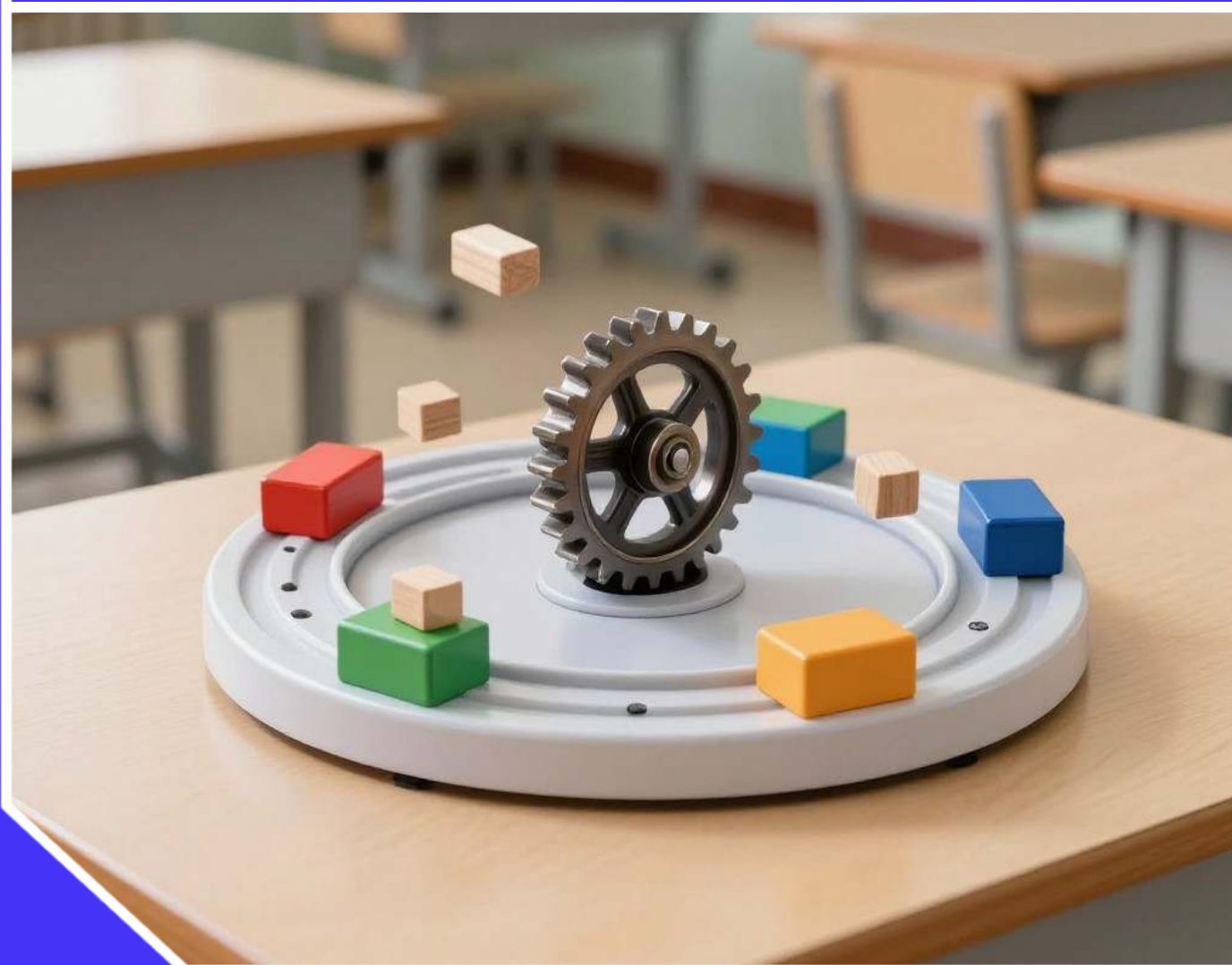


Циклические алгоритмы: повторение действий в информатике

Цикл помогает описать повторяющиеся шаги один раз
и выполнить их нужное число раз.





В 7 классе циклы встречаются в первых алгоритмах и программах: считаем, ищем, печатаем последовательности и строим рисунки. Они сокращают запись одинаковых команд, но требуют точного числа повторов, правильного условия завершения и изменения переменной.

Почему циклы особенно важны в 7 классе

Что такое циклический алгоритм

01

Циклический алгоритм выполняет один и тот же набор команд несколько раз. Каждый повтор называется итерацией, а общая структура помогает не переписывать одинаковые шаги снова и снова.

02

В цикле есть тело цикла — команды, которые повторяются, и правило продолжения. Именно оно решает, будут ли выполняться команды ещё раз или работа прекратится.

03

Если правило задано неверно, программа может тратить лишнее время или вообще не завершиться. Поэтому важно продумать логику остановки до начала выполнения.

Три признака цикла

- Первый признак — повторяющиеся действия: одна и та же операция входит в алгоритм не один раз. Это отличает цикл от обычной последовательности команд.
- Второй признак — условие или счётчик, которые определяют, когда повторение заканчивается. Благодаря этому программа знает, сколько раз выполнить тело или когда остановиться.
- Третий признак — остановка, после которой работа завершается. Без корректного завершения получается бесконечный цикл, а без изменения счётчика повтор может идти неправильно.

Сколько раз повторяются действия в разных задачах



Примерные значения из учебных ситуаций (разминка, упражнения, числа, месяцы, тренировка); точные данные в учебнике могут отличаться.

Даже при небольшом числе повторов цикл делает алгоритм короче. При росте числа шагов экономия становится заметнее.

Диаграмма показывает: цикл полезен и для 5–7 повторов, и для 20 повторов. Чем больше однотипных действий, тем меньше ручной записи требуется.

Цикл со счётчиком

- Цикл со счётчиком используют, когда количество повторений известно заранее. Он помогает выполнить тело ровно заданное число раз, не проверяя условие каждый шаг.
- Счётчик последовательно принимает значения, например от 1 до 10 или от 0 до 5. На каждом проходе он меняется, чаще всего на 1, чтобы число повторов не «зависло».
- Такой цикл удобен для вывода таблиц, подсчёта элементов и печати последовательностей. Важно правильно выбрать начальное значение и верхний предел счётчика.

ОСНОВНЫЕ ВИДЫ ЦИКЛОВ

Сравним три конструкции: когда применяется каждая, что проверяется и как устроено повторение.

Выбор цикла зависит от момента проверки: до тела, после тела или от фиксированного числа повторов. Это влияет на то, будет ли тело выполнено ноль раз.

Вид цикла	Когда нужен	Что проверяется	Проход минимум	Пример
Со счётчиком	Число повторов заранее известно	Счётчик и предел	Не менее 0	Вывод 1..10
С условием	Повторять, пока условие верно	Условие до входа в тело	Не менее 0	Пока число < 10
С постусловием	Действие нужно хотя бы раз	Условие после тела	Хотя бы 1 раз	Ввод до верного ответа

Учебный курс информатики для 7 класса (обобщение типовых определений).

Цикл с условием в повседневной ситуации

Смысл повторения

Цикл с условием повторяется, пока выполняется нужное правило. Как только условие перестаёт быть верным, программа прекращает повторять команды и переходит к следующему действию.

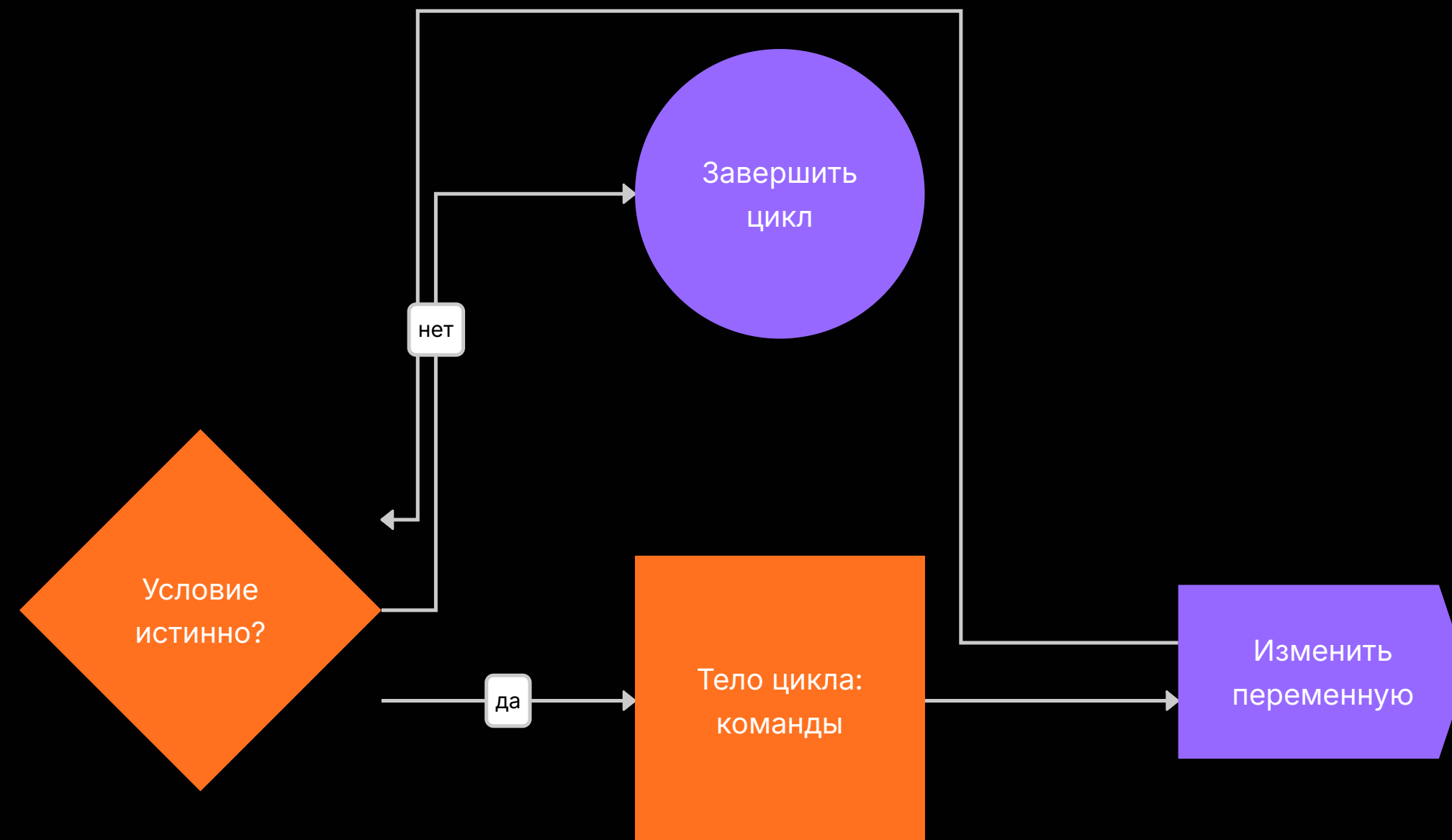
Жизненный пример

Например, «идти через турникет, пока проход открыт» или «шаги прибавляются, пока включён счётчик». Движение продолжается, пока условие остаётся истинным, и останавливается при смене состояния.



Как работает цикл с условием

Повторение происходит по шагам: проверка → выполнение → изменение → снова проверка.



Цикл с постусловием

01

Цикл с постусловием сначала выполняет команды, а затем проверяет условие продолжения. Поэтому тело цикла стартует сразу, без предварительной проверки.

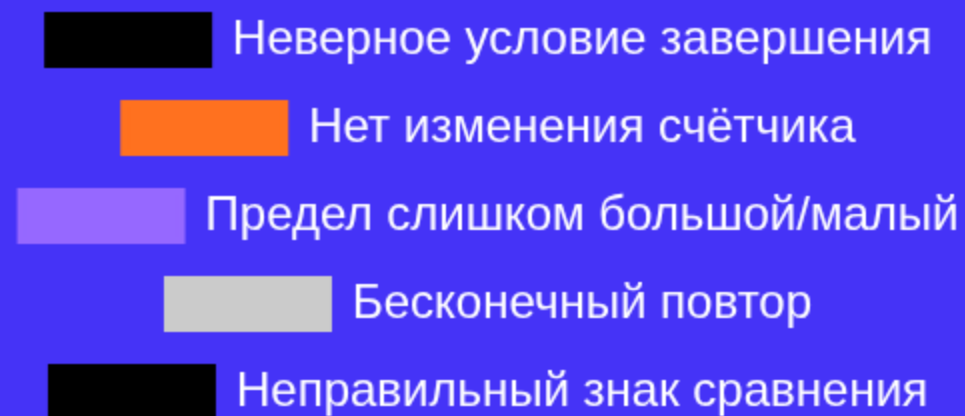
02

Такой подход подходит, когда действие нужно сделать хотя бы один раз. В учебных задачах это может быть повторная попытка ввода или проверка данных после первой попытки.

03

Главное отличие — проверка стоит после тела цикла, а не перед ним. Благодаря этому можно не пропустить первый запуск, даже если условие окажется ложным уже после выполнения.

Типичные ошибки при работе с циклами



Учебное описание ошибок (качественная классификация), без статистических данных; 2026-05-14

В задачах по циклам проблема часто не в теле цикла, а в «правилах остановки» и изменении параметров. Проверь условие, границы и шаг до того, как запускать программу.

Круговая диаграмма показывает, что самые частые сбои связаны с остановкой и счётчиком. Если цикл не приходит к завершению, время растёт или выполнение зависает.

Бесконечный цикл и способы его избежать

- Бесконечный цикл получается, когда условие продолжения никогда не станет ложным. Часто виноваты забытое изменение переменной или шаг, который не ведёт к нужному пределу, из-за чего цикл не завершается.
- Ещё одна причина — ошибка сравнения: знак «меньше» заменили на «больше», либо перепутали «равно» и «не равно». Тогда условие может оставаться истинным слишком долго или вообще не выполняться нужное число раз.
- Чтобы поймать ошибку заранее, полезно «прогнать» алгоритм по шагам: выписать начальные значения, затем каждую проверку условия и обновление счётчика. Так сбой виден ещё до запуска программы.

Примеры циклов в школьных задачах

В таблице показано, какое действие повторяется и как цикл понимает, что пора остановиться. Такие формулировки помогают выбрать правильный вид цикла в задании.

Во всех примерах цикл удобен, потому что есть повторяемое действие и понятный критерий завершения. Это делает алгоритм короче и уменьшает риск забыть очередной шаг.

Задача	Повторяющееся действие и остановка
От 1 до 10	Печать очередного числа; остановка при достижении 10
Сложить оценки	Добавление очередной оценки; остановка после последней оценки
Найти максимум	Сравнение текущего значения с лучшим; остановка после конца списка
Квадрат из символов	Печать строк и столбцов символов; остановка после нужного размера
Повторить движение 8 раз	Повторять команду движения; остановка после 8 повторов

Типовые школьные задания по информатике (7 класс), на основе описания в учебном курсе.

Циклы в играх и приложениях

Один кадр — много обновлений

На экране игры персонажи смещаются почти незаметно, потому что изображение обновляется снова и снова. В каждом цикле пересчитывают положение объектов, проверяют события и готовят кадр к следующему показу. Так движение получается плавным, а мир реагирует на действия игрока.



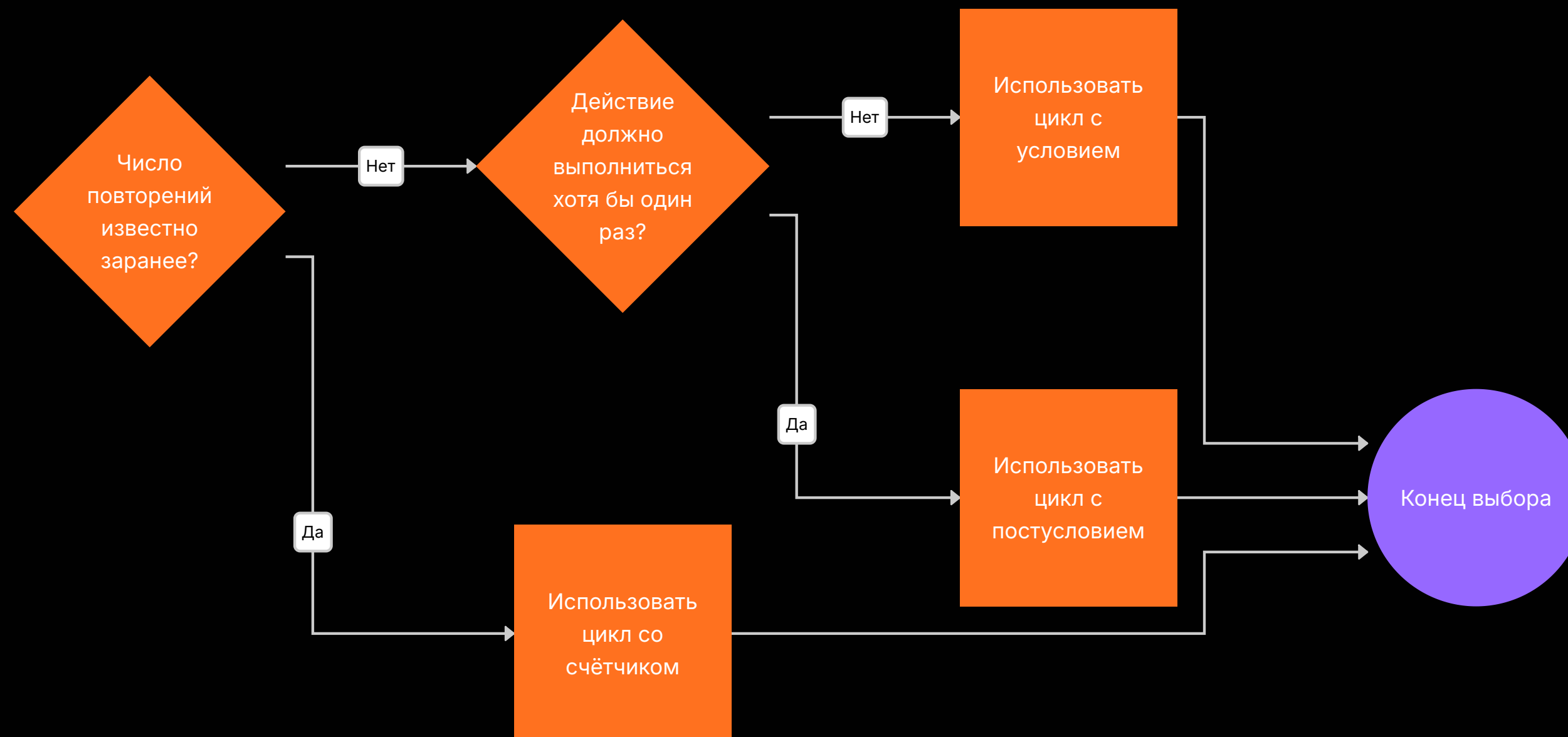
Время, ввод и новые сообщения

В приложениях цикл помогает не «замирать» между действиями пользователя. Он регулярно проверяет, прошло ли время, пришёл ли ввод или появилось ли новое сообщение. Благодаря повторной проверке интерфейс остаётся отзывчивым и быстро реагирует на изменения без ручного повторения команд.



Как выбрать подходящий вид цикла

Схема выбора: известны ли заранее повторы и должно ли действие выполниться минимум один раз.



Циклы в псевдокоде и программах



Структура цикла

В записи цикла обычно есть заголовок, условие и блок команд. Псевдокод помогает увидеть эти части без привязки к конкретному языку, чтобы легче анализировать логику.



Условие продолжения

Перед разбором полезно определить, что считается правильным для продолжения повторов. Тогда легко понять, почему цикл завершится или, наоборот, рискует стать бесконечным при неверном сравнении.



Повтор до результата

Смысл цикла одинаков в разных языках: повторять действия, пока не достигнут нужный результат или предел. Это позволяет формулировать задачу через «что проверяем» и «когда останавливаемся».



Проверка алгоритма

В 7 классе удобно мысленно выполнять псевдокод по шагам: выписать значения счётчика, затем проверять условие и обновлять переменные. Так находят ошибки быстрее, чем по готовому коду.

Изменение значения при повторении



Пример из учебной темы (последовательность 2, 4, 6, 8, 10), 2026-05-14

Линейная линия показывает правило: при каждом новом повторе значение увеличивается на 2. Такой вид графика легко связать с работой счётчика и фиксированным шагом.

По данным видно, что результат растёт предсказуемо и равномерно. Это типичный сценарий для циклов со счётчиком, когда шаг и предел заданы заранее.

Почему ЦИКЛЫ ЭКОНОМЯТ время

- Один цикл заменяет множество одинаковых команд. Благодаря этому программа получается короче, её проще проверить и легче исправить: ошибка в одной команде заменяется корректировкой правила повторения.

- При больших наборах данных ручной ввод повторов занимает слишком много времени. Циклы позволяют автоматически обработать список значений, выполнив одно и то же действие для каждого элемента по порядку.

- В учебных задачах циклы быстро дают таблицы, последовательности чисел и повторяющиеся рисунки из символов. Вместо десятков строк достаточно описать правило и критерий окончания повторов.

Ключевые понятия темы

Мини-словарь помогает не путать термины и быстрее разбирать алгоритмы с циклами. Формулировки короткие и связаны с типичными школьными заданиями.

Таблица показывает, что понимание термина напрямую связано с тем, как устроена остановка. Когда известны цикл, тело, итерации, счётчик и условие, ошибки предсказать легче.

Термин	Коротко и с примером
Цикл	Повторяет действия по правилу; пример: «повторить 8 раз»
Тело цикла	Команды, которые выполняются в каждом повторе; пример: «напечатать символ»
Итерация	Один проход выполнения тела цикла; пример: «очередная печать строки»
Счётчик	Переменная, задающая номер повторения; пример: от 1 до 10
Условие	Проверка, продолжать или остановиться; пример: «пока число меньше 10»
Бесконечный цикл	Условие никогда не становится ложным; пример: счётчик не меняется
Постусловие	Проверка после тела цикла; пример: действие выполняется хотя бы один раз

Учебный курс информатики для 7 класса: определения ключевых понятий по теме циклов.



Циклический алгоритм повторяет действия по правилу и делает запись короче. Выбор вида цикла зависит от того, известны ли повторы и нужно ли выполнять действие хотя бы один раз. Условие и правильная остановка критичны.

Главные выводы по теме циклических алгоритмов