



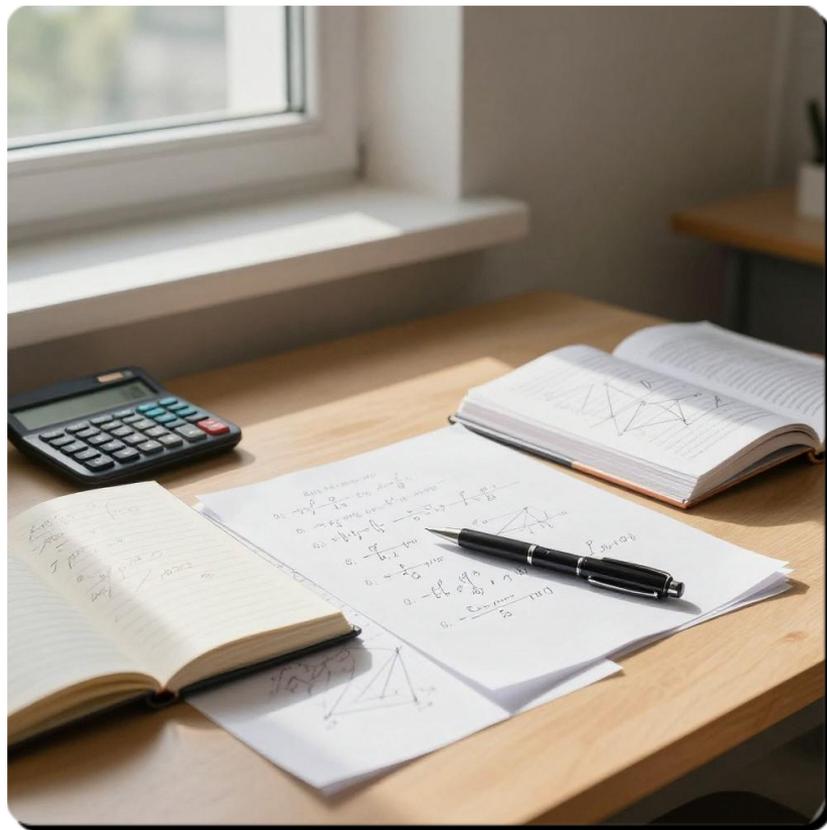
Формирование гибкого мышления: аналитические и алгоритмические подходы к решению задач ЕГЭ

2026 г

Мальцева Ирина Анатольевна
Региональный методист,
учитель информатики MAOY COШ №92 города Тюмени

Изменения в оценивании ЕГЭ: взгляд в будущее

Современный ЕГЭ смещает фокус с заучивания на умение применять знания в новых условиях и разбираться в сложных задачах, требующих функциональной грамотности.



01

Гибкое мышление — способность переключаться между разными методами решения в зависимости от задачи, а не ограничиваться единственным алгоритмом.

02

Важна умственная адаптация: модификация традиционных алгоритмов под новые или изменённые условия задач, ответы на нестандартные вызовы.

03

Контроль собственного решения — отслеживание логики хода мысли, проверка промежуточных результатов и осознанный выбор стратегии.

04

Видение задачи с различных ракурсов, а также использование метакогнитивных навыков для анализа ситуации.

Что такое гибкое мышление на ЕГЭ?



Проблемы ригидного мышления: распространённые ошибки

При столкновении с незнакомыми или изменёнными условиями экзаменационных заданий учащиеся часто оказываются неспособны адаптировать решения, что приводит к параличу мыслей.

Слепое следование формулам без понимания сути задачи часто становится причиной грубых ошибок и снижает итоговый балл, особенно в заданиях с развёрнутым ответом.

Алгоритмический подход (фундамент уверенности): плюсы и минусы



Обеспечивает структурированность решений, повышает скорость и уверенность при решении стандартных заданий базового уровня.



Снижает тревожность за счёт чёткого плана, позволяя действовать по шаблону и экономить умственные ресурсы.



Однако высокая зависимость от стандартных схем уменьшает гибкость и затрудняет адаптацию к новым типам задач.



Для эффективности необходима целенаправленная отработка шаблонных задач и формирование чек-листов для быстрой проверки решений.

Аналитический подход (инструмент для высоких баллов)

Направлен на понимание сути явления, выявление связей и построение уникальной логики решения.

Разбиение сложной задачи на подзадачи.
Понимание математического смысла ответа до вычислений.

Поиск альтернатив:
Вопрос «А можно ли решить это другим способом?».

Заставляют анализировать условие.
Метод «Обратной задачи»: Дано решение — восстановите условие.
Анализ ошибок:
Разбор не только того, как надо было решить, но и почему возникла ошибка в логике.

Синтез подходов: формула гибкости

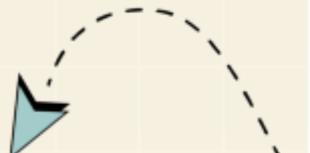
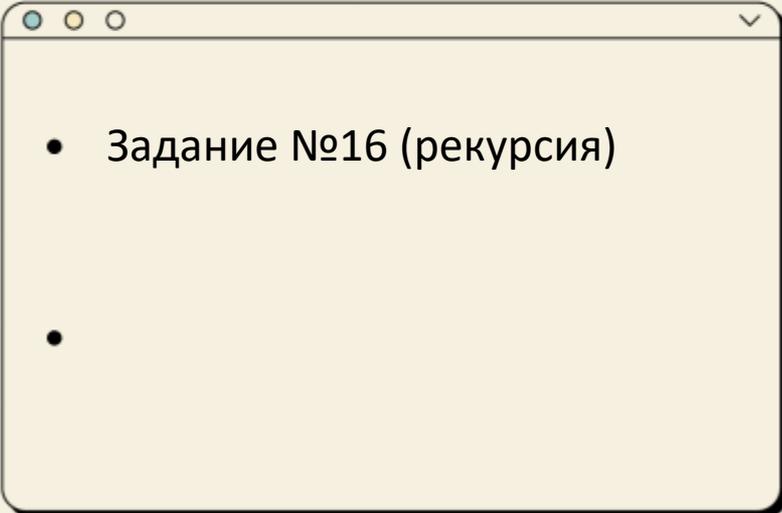
Алгоритм и Анализ не исключают, а дополняют друг друга.

Гибкое мышление формируется на стыке этих подходов по следующей схеме:

Попытка алгоритма: Ученик сканирует задачу на наличие знакомого типа.

Аналитическая пауза: Если алгоритм не подходит, включается анализ («Почему не работает? Что здесь нового?»).

Модификация: Ученик адаптирует известный алгоритм или комбинирует несколько методов.

- 
- 
- Задание №16 (рекурсия)

-

Задача: № 25399 (Взаиморекурсивные функции с большими числами).

Ученик видит рекурсивные формулы и действует по стандартному шаблону: «Видишь рекурсию — пиши функцию, вызывающую саму себя».

```
def F(n):
    if n >= 128:
        return F(n - 5) + 1092
    else:
        return 5 * G(n - 7) + 29
def G(n):
    if n > 303728:
        return n - 15
    else:
        return G(n + 8) / 2 - 109
print(F(2049))
```

Ожидание: Программа посчитает ответ.

Реальность: Программа падает с ошибкой *RecursionError*.

Препятствие (Анализ причин)

Почему возникла ошибка? Ученик с гибким мышлением не просто «увеличивает лимит рекурсии», а анализирует глубину вызовов:

Задача: № 25399 (Взаиморекурсивные функции с большими числами).

Функция F: вызывается с шагом -5 от 2049 до 124. Это ~385 вызовов. (Нормально).
Функция G: вызывается из F(124) как G(117).
Условие выхода: $n > 303728$
Шаг: +8.

Глубина:
 $(303728 - 117) / 8 \approx 38\,000$
 $(303728 - 117) / 8 \approx 38\,000$
вызовов.
Проблема: стандартный лимит стека в Python — 1000. Превышение в 38 раз.

Гибкое мышление:
«Рекурсия здесь не подходит по ресурсам. Нужно менять стратегию вычисления».

1. Математика:
Вычисляет точку входа в базу для G (число 303733) и точку выхода из F (число 124) через операцию остатка от деления (%).
2. Алгоритм: Заменяет рекурсивную функцию G на **цикл**. Цикл не имеет ограничения по глубине стека и легко обрабатывает 38 000 итераций за доли секунды.

Задача: № 25399 (Взаиморекурсивные функции с большими числами).

1.py - C:/Users/Malzeva/AppData/Local/Programs/Python/Python312/Lib/idlelib/1.py (3.12.7)

File Edit Format Run Options Window Help

```
# Гибкое решение
# 1. Считаем F аналитически до точки перехода
steps_F = (2049 - 124) // 5
val_F = 385 * 1092 # Накопленная сумма

# 2. Считаем G итеративно (циклом), а не рекурсивно
# Находим точку старта для G (первое число > 303728, дающее остаток 5 при делении на 8)
start_G = 303733
val_G = start_G - 15 # Базовое значение

# Спускаемся вниз циклом (38000 раз)
for _ in range(37952):
    val_G = val_G / 2 - 109

# 3. Собираем ответ
answer = val_F + (5 * val_G + 29)
print(answer)
```

IDLE Shell 3.12.7

File Edit Shell Debug Options Window Help

```
Python 3.12.7 (tags/v3.12.7:0b05ead, Oct 1 2024, 03:06:41) [MSC v.1941 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Malzeva/AppData/Local/Programs/Python/Python312/Lib/idlelib/1.py
419359.0
>>>
```

Задача: № 25399 (Взаиморекурсивные функции с большими числами).

Краткое решение (для проверки):

1. Упрощение $F(2049)$:

- Функция $F(n)$ уменьшает аргумент на 5, пока $n \geq 128$.
- Нам нужно найти первое число < 128 , которое получится из 2049 вычитанием пятерок.
- $2049 \pmod{5} = 4$. Ближайшее число < 128 с остатком 4 — это 124.
- Количество шагов: $(2049 - 124)/5 = 385$.
- Накопленная сумма: $385 \times 1092 = 420420$.
- Получаем: $F(2049) = F(124) + 420420$.

2. Вычисление $F(124)$:

- Так как $124 < 128$, срабатывает формула: $F(124) = 5 \times G(117) + 29$.
- Нужно найти $G(117)$.

3. Вычисление $G(117)$ (Аналитический инсайт):

- Функция $G(n)$ увеличивает аргумент на 8, пока $n \leq 303728$.
- Это более 38 000 шагов. Писать рекурсию или цикл долго и рискованно.
- Замечаем, что формула $G(n) = G(n + 8)/2 - 109$ — это сходящийся процесс.
- Найдем предел (точку равновесия), где значение перестает меняться: $x = x/2 - 109 \Rightarrow x/2 = -109 \Rightarrow x = -218$.
- Поскольку шагов очень много (37 952), начальное значение «забывается», и функция стремится к -218.
- Принимаем $G(117) = -218$.

4. Финальный подсчет:

- $F(124) = 5 \times (-218) + 29 = -1090 + 29 = -1061$.
- $F(2049) = -1061 + 420420 = \mathbf{419359}$.



Этот пример идеально иллюстрирует **гибкость мышления**:

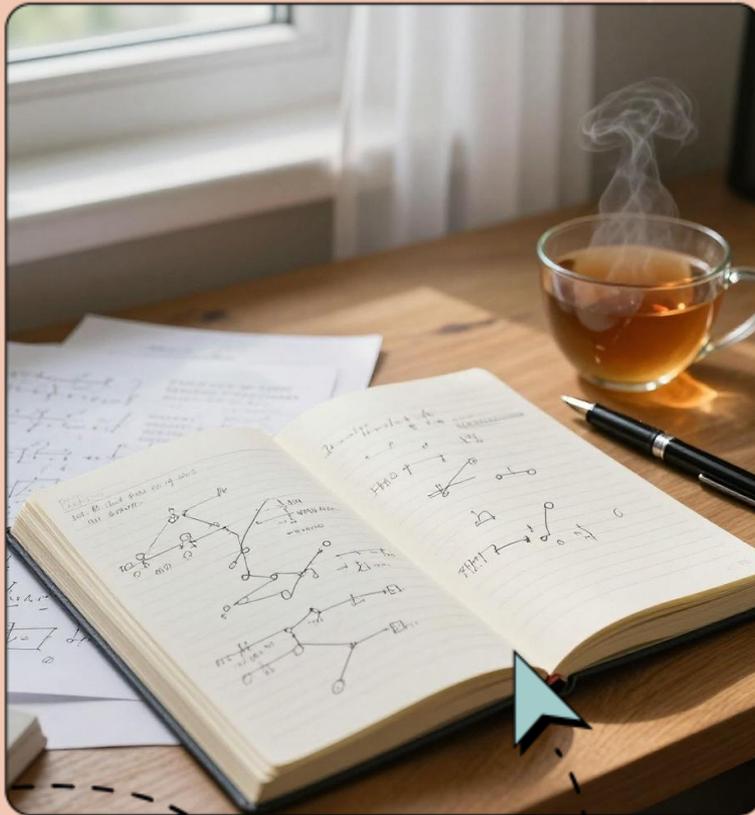
•**Ригидный подход:**

Пытаться написать рекурсивную функцию → получить ошибку переполнения стека ([RecursionError](#)).

•**Гибкий подход:**

Проанализировать математику процесса, заметить сходимость к пределу и заменить 38 000 итераций одним числом (-218). Это экономит время и исключает ошибки вычислений.

Ключевые навыки аналитического подхода



Декомпозиция и альтернативные подходы

Аналитический подход включает разбиение задачи на части, поиск альтернативных путей решения и умение предварительно оценить результат для контроля адекватности.

Оценка решения и анализ ошибок

Для развития мышления важен системный разбор причин ошибок, а также работа с задачами, где данные неполные или избыточные, стимулируя глубокий анализ.

Сравнение алгоритмического и аналитического методов

- Основные характеристики методов решения, выявленные на основе анализа результатов ЕГЭ 2024-2025 годов.

- Аналитический метод обеспечивает большую адаптивность и точность, в то время как алгоритмический ускоряет базовые решения.

Критерий	Алгоритмический подход	Аналитический подход
Скорость решения	Высокая	Умеренная
Устойчивость к новизне	Низкая	Высокая
Адаптивность	Ограниченная	Широкая
Глубина понимания	Низкая	Высокая
Частота ошибок	Выше при нестандартных задачах	Ниже

Практические методики для урока

Чтобы реализовать это на практике, предлагаю использовать следующие приемы:

«Три пути»: Предлагать одну задачу и требовать найти минимум два разных способа её решения. Это ломает привязку к единственному алгоритму.

«Намеренная ошибка»: Учитель решает задачу на доске с ошибкой в логике. Задача учеников — найти и аргументировать, где мышление пошло не туда.

Тайм-менеджмент мышления: Учить детей правилу «5 минут». Если задача не решается стандартным методом за 5 минут — принудительная смена стратегии или переход к другой задаче (тренировка гибкости переключения).

Визуализация: Требовать обязательного построения графика, чертежа или схемы даже в тех задачах, где это не очевидно. Это переводит абстрактное в наглядное.

Психологический аспект

Важно помнить, что гибкость мышления блокируется стрессом.

В состоянии паники мозг переходит на самые примитивные, ригидные реакции.

Поэтому тренировка гибкости должна проходить в условиях, имитирующих экзаменационный стресс, но с постепенным повышением нагрузки.

Необходимо формировать установку: *«Трудная задача — это не тупик, это повод сменить инструмент».*

Заключение: синергия аналитики и алгоритмов как ключ к успеху



Коллеги, подготовка к ЕГЭ — это не натаскивание на тесты. Это развитие мышления.

Алгоритмический подход дает ученику уверенность и базу.

Аналитический подход дает возможность преодолеть нестандартные препятствия. Их сочетание рождает **гибкое мышление**.

Наша задача — научить ребенка не бояться неизвестного, а исследовать его, используя имеющийся инструментарий. Это навык, который пригодится им не только на экзамене, но и в дальнейшей учебе, и в профессиональной деятельности.





Спасибо за внимание!

