Тестирование программного обеспечения

Тестирование - один из трудоемких процессов.

Существует ряд методов контроля и тестирования результатов.

Нужно решить задачу, связанную:
с выбором стратегии и методов тестирования разработке сложных программных систем.

В частности, при подготовке к тестированию необходимо ответить на следующие вопросы:

- Какую стратегию тестирования выбрать и почему?
- Как реализовать стратегию тестирования?
- Какой из методов стратегии тестирования выбрать и почему?
- Как грамотно подготовить тестовый набор данных и сколько тестов необходимо разработать?
- Как выполнить эффективное тестирование?

Основные принципы тестирования

Тестированием называется процесс выполнения программы ***с целью обнаружения ошибки***.

Необходимо отметить, что никакое тестирование не может доказать отсутствие ошибок в программе.

При тестировании **рекомендуется соблюдать** следующие *основные принципы (см. методичку).*

- Ожидаемые результаты должны быть определены до тестирования;

- Необходимо проверять действия программы на неверных данных;

- Вероятность наличия ошибки в части программы пропорциональна количеству ошибок, уже обнаруженных в этой части и др.

Ручное тестирование

Эксперименты показали, что это эффективный метод.

Методы ручного контроля предназначены для периода разработки, **когда программа закодирована**, но тестирование на машине еще не началось.

Основными методами ручного тестирования являются:

- инспекции исходного текста;
- сквозные просмотры;
- просмотры за столом;
- обзоры программ.

Инспекции исходного текста

Состоит из следующих шагов:

- участникам группы заранее **выдается листинг программы и спецификация** на нее;

- программист рассказывает о логике работы программы и отвечает на вопросы инспекторов;

- программа анализируется **по списку вопросов** для выявления исторически сложившихся общих ошибок программирования.

Сквозные просмотры

Имеет много общего с процессом инспектирования.

Отличается процедурой и методами обнаружения ошибок.

Состоит из следующих шагов:

- участникам группы заранее выдается листинг программы и спецификация на нее;
- участникам заседания предлагается несколько тестов, написанных на бумаге (каждый тест мысленно выполняется);
- программисту задаются вопросы о логике проектирования и принятых допущениях;
- состояние программы (значения переменных) отслеживается на бумаге или доске.

Проверка за столом

Используется ранее других методов.

Включает **проверку исходного кода** программы (или сквозной просмотр), выполняемую **одним человеком**.

Проверка **по списку вопросов** и пропускает через программу тестовые данные.

Должен проводить человек, который **не является автором** программы.

Оценка посредством просмотра

Цель метода - обеспечить сравнительно объективную оценку и самооценку программистов.

- ✓ Выбирается программист - администратор процесса.
- ✓ Администратор набирает группу от 6 до 20 участников.
- ✓ Каждому участнику предлагается выбрать наилучшую и наихудшую программу.
- ✓ Отобранные программы случайным образом распределяются между участниками.
- ✓ Участникам дается по 4 программы - две наилучшие и две наихудшие (программист про это не знает)ю
- ✓ Программист оценивает программы по семи шкале.
- ✓ Проверяющий дает общий комментарий и рекомендации по улучшению программы.

Вопросы для структурного контроля текста

1. *Обращения к данным.*

1. Все ли переменные инициализированы?
2. Не превышены ли максимальные (или реальные) размеры массивов и строк? и др.

2. *Вычисления.*

1. Правильно ли записаны выражения (порядок следования операторов)?
2. Корректно ли производятся вычисления неарифметических переменных? и др.

3. *Передачи управления.*

1. Будут ли корректно завершены циклы?
2. Будет ли завершена программа? и др.

4.*Интерфейс.*

1. Соответствуют ли списки параметров и аргументов по порядку, типу, единицам измерения?
2. Не изменяет ли подпрограмма аргументов, которые не должны изменяться? и др.

Тестирование по принципу «белого ящика»

Позволяет проверить внутреннюю структуру программы.

Проверка проведена полностью, если с помощью тестов удалось проверить все возможные маршруты передачи управления.

Но даже в программе среднего уровня сложности число неповторяющихся маршрутов очень большое.

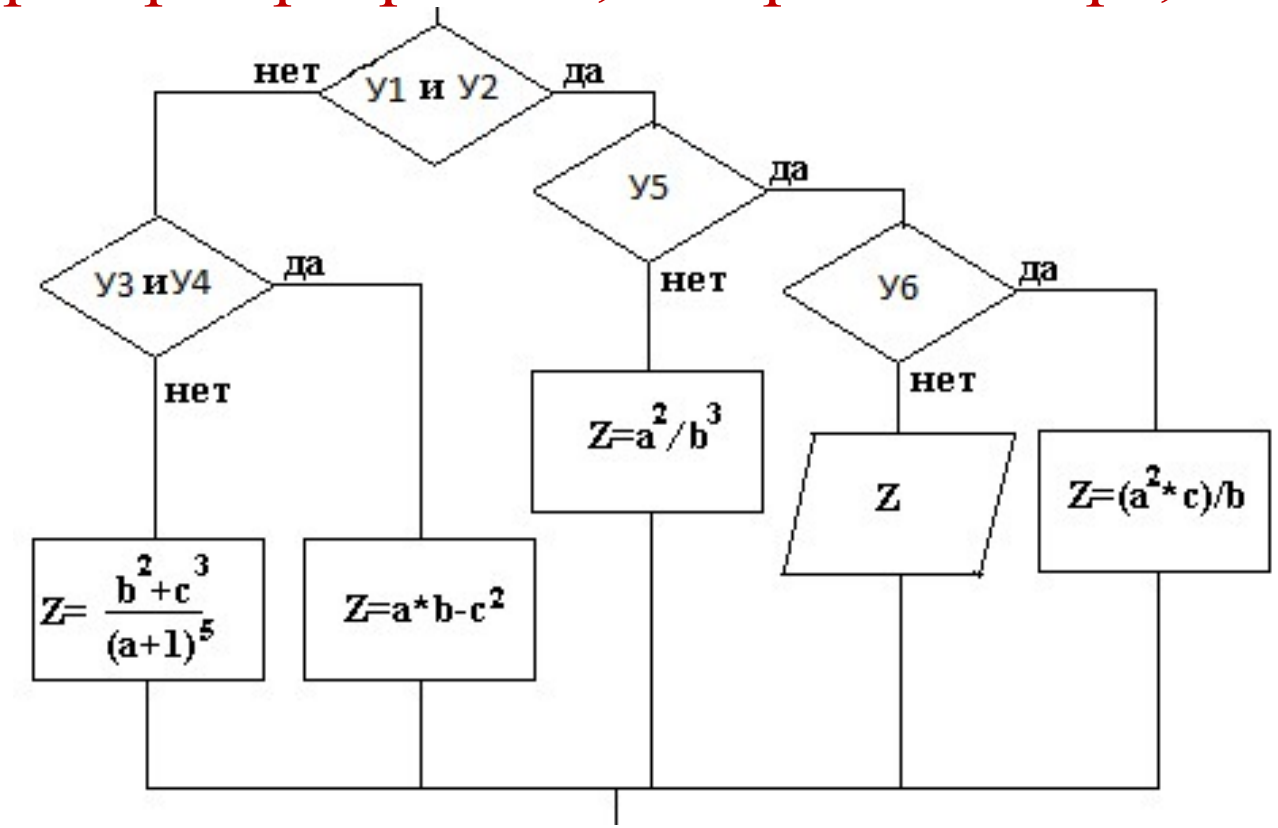
Стратегия «белого ящика» включает в себя следующие методы тестирования:

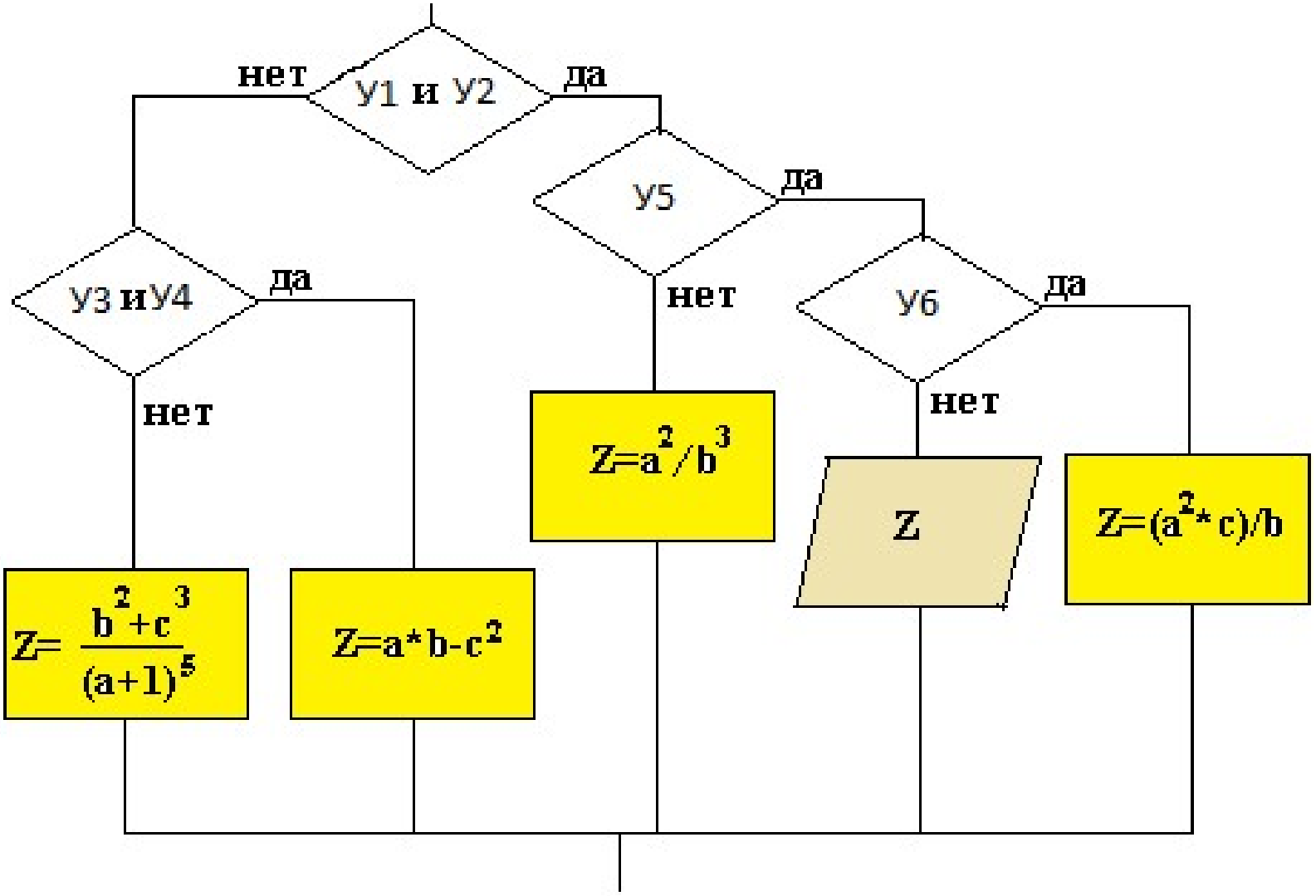
- Покрытие операторов.
- Покрытие решений.
- Покрытие условий.
- Покрытие решений/условий.
- Комбинаторное покрытие условий.

Рассмотрим некоторые из них.

Покрытие операторов

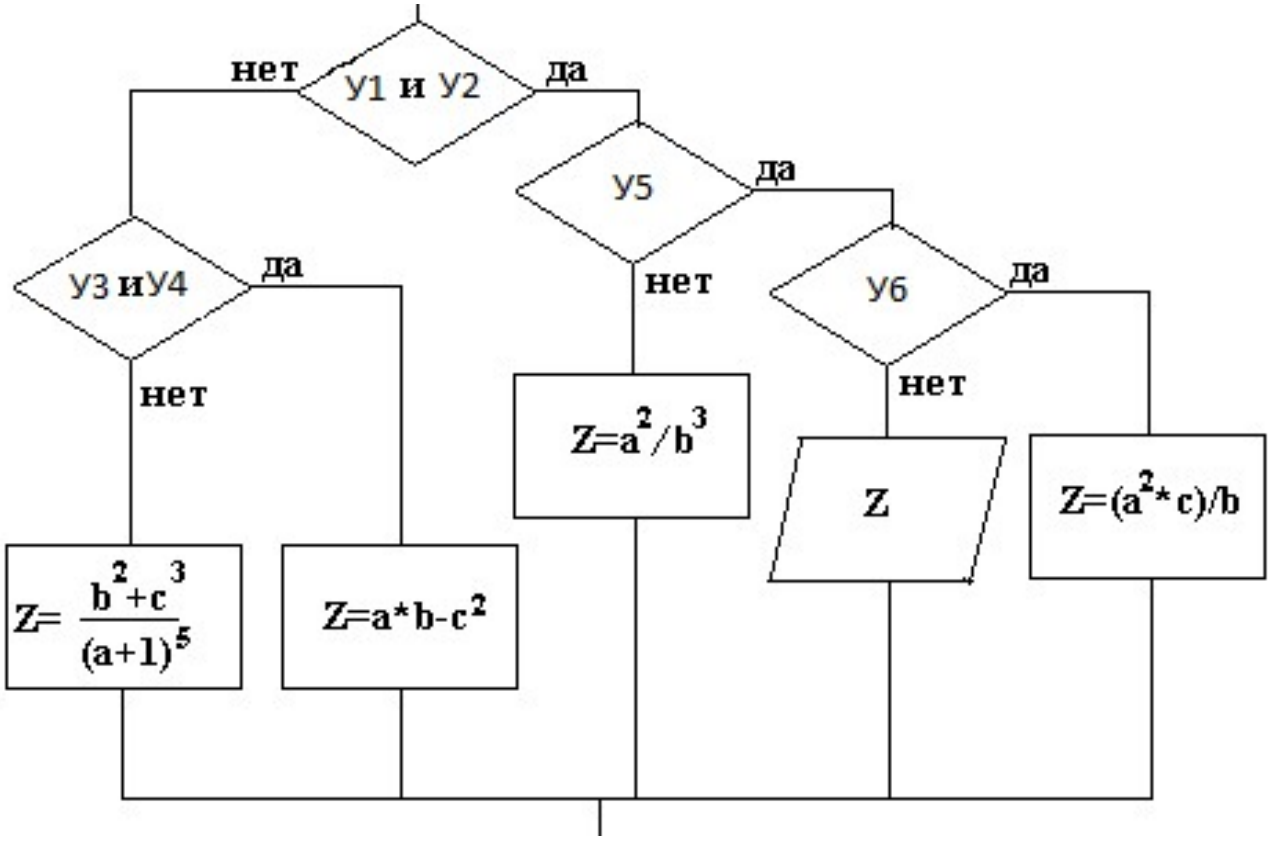
Критерий покрытия операторов подразумевает выполнение каждого оператора программы, по крайней мере, один раз.

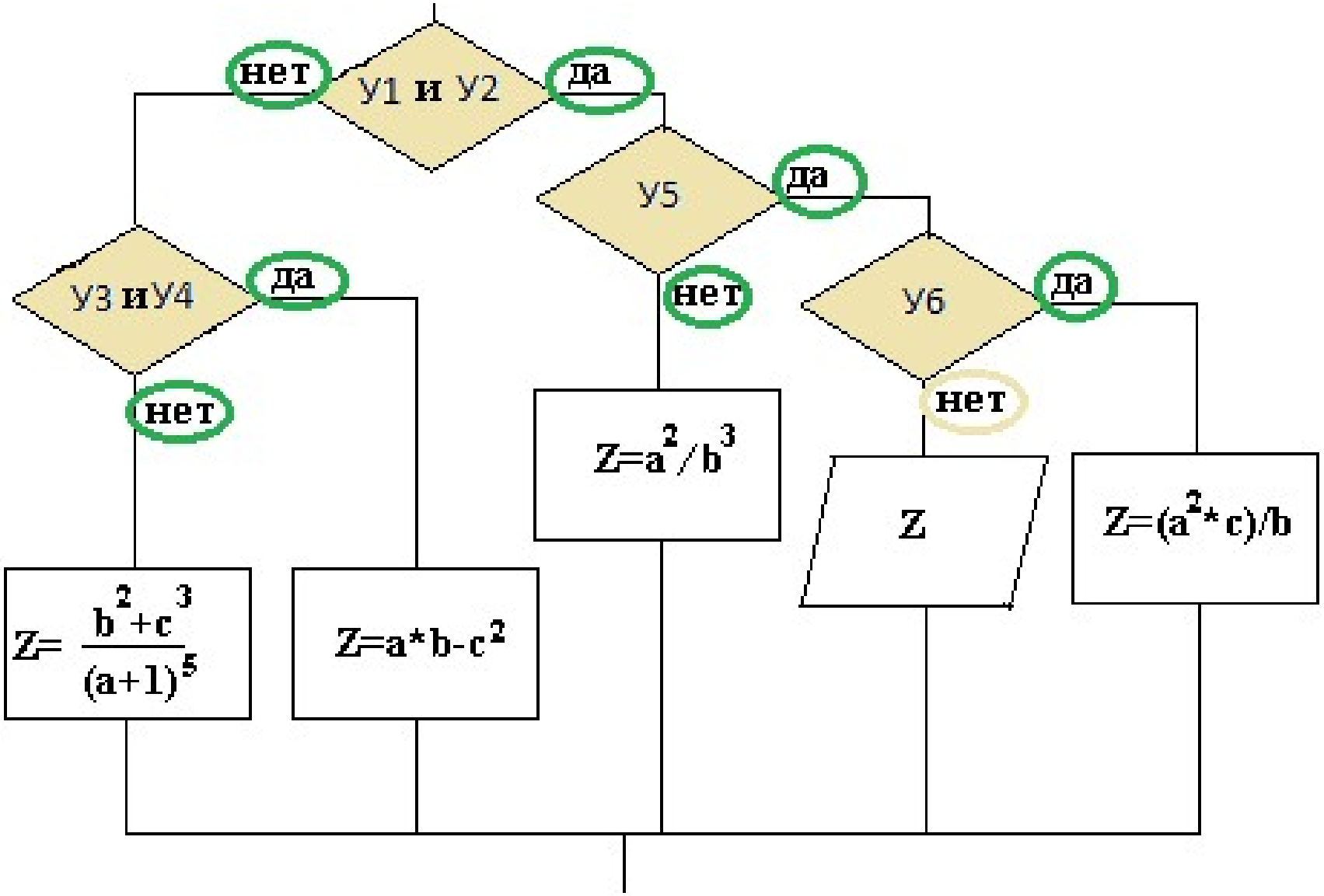



У1 и У2
нет
да
У5
да
нет
У3 иУ4
да
нет
У6
да
нет
Z=a²/b³
Z
Z=(a²*c)/b
Z= (b²+c³)/(a+1)⁵
Z=a*b-c²

Покрытие решений (переходов)

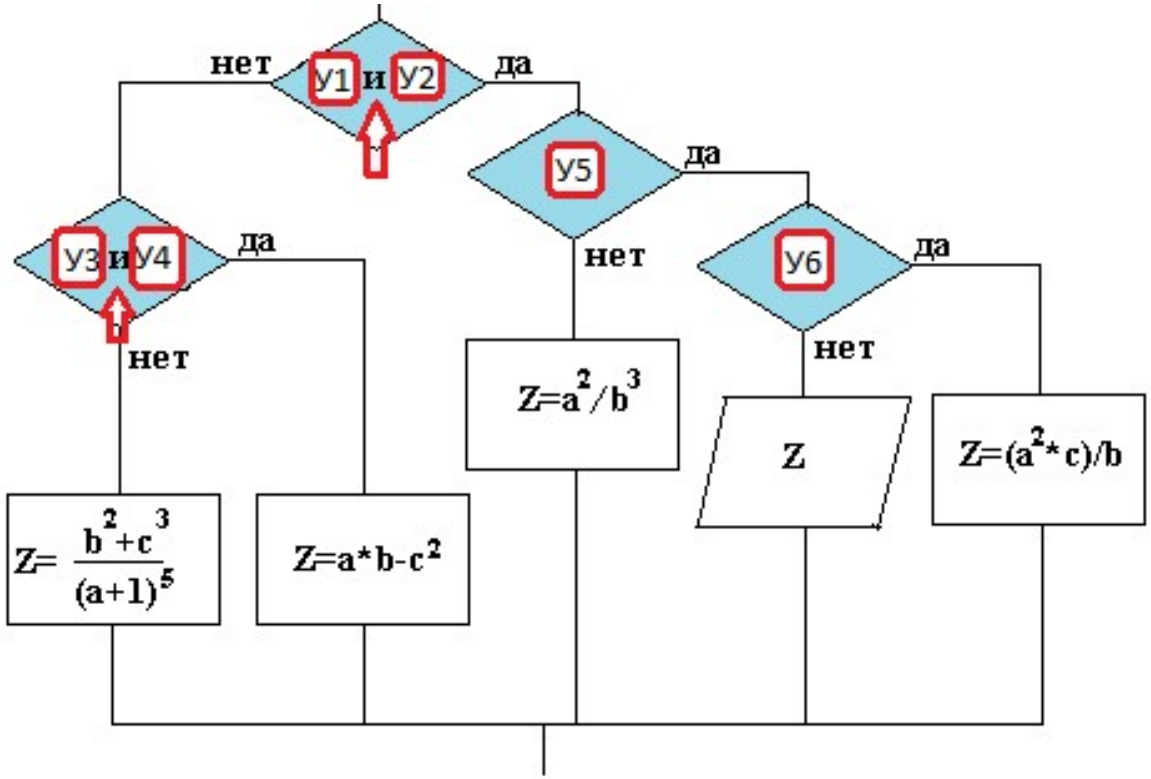
Каждое решение нужно покрыть, по крайней мере, один раз.




нет
У1 и У2
да
У5
да
У3 иУ4
да
нет
нет
У6
да
нет
$Z=\frac{b^2+c^3}{(a+1)^5}$
$Z=a*b-c^2$
$Z=a^2/b^3$
Z
$Z=(a^2*c)/b$

Комбинаторное покрытие условий

Этот критерий требует создания такого числа тестов, чтобы все возможные комбинации результатов условий в каждом решении и все точки входа выполнялись, по крайней мере, один раз.



Тестирование по принципу «черного ящика»

Это тестирование с управлением по данным.

Программа рассматривается как "черный ящик".

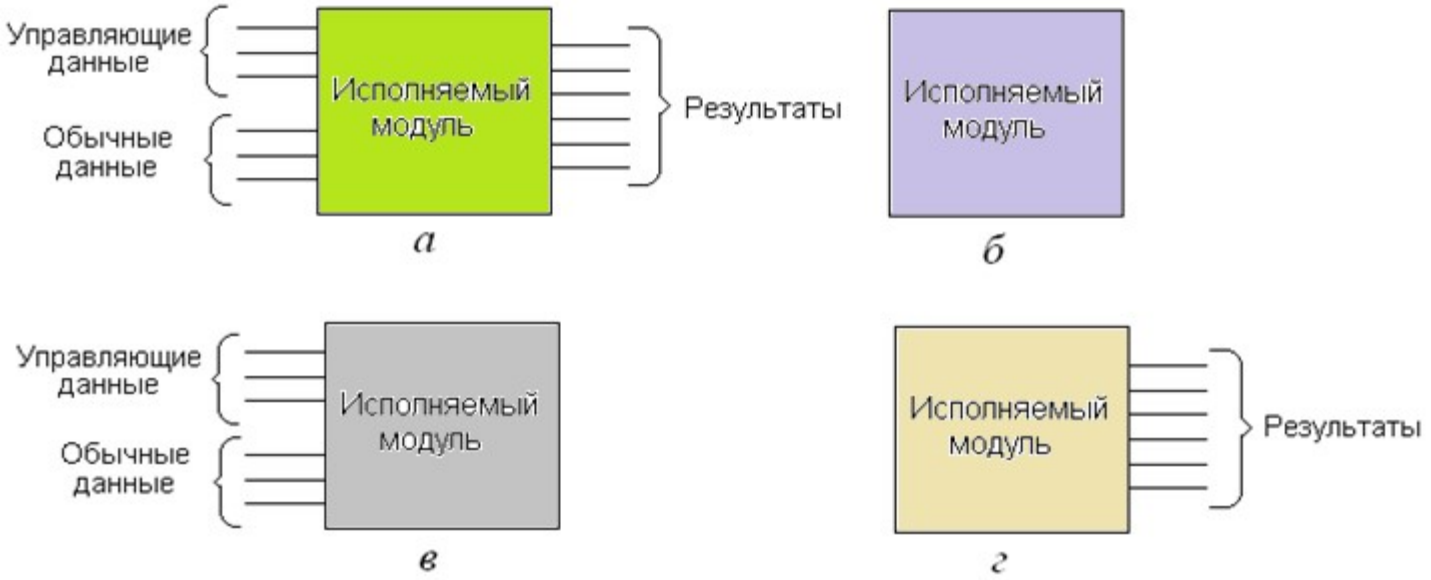
Цель - выяснить на соответствие спецификации.

Если исполнение команды зависит от предшествующих ей событий, то необходимо проверить все возможные последовательности. **Их может быть много.**

Поэтому, обычно выполняется ***"разумное" тестирование***.

Стратегия "черного ящика" включает

- эквивалентное разбиение;
- анализ граничных значений;
- анализ причинно-следственных связей;
- предположение об ошибке.



Эквивалентное разбиение

Разработка тестов осуществляется в два этапа:

- ✓ выделение классов эквивалентности;
- ✓ построение тестов.

Анализ граничных значений

Граничные условия - это ситуации, возникающие вблизи и на границах входных классов эквивалентности.

Анализ причинно-следственных связей

Необходимо понимание булевской логики (логических операторов - и, или, не).

Предположение об ошибке

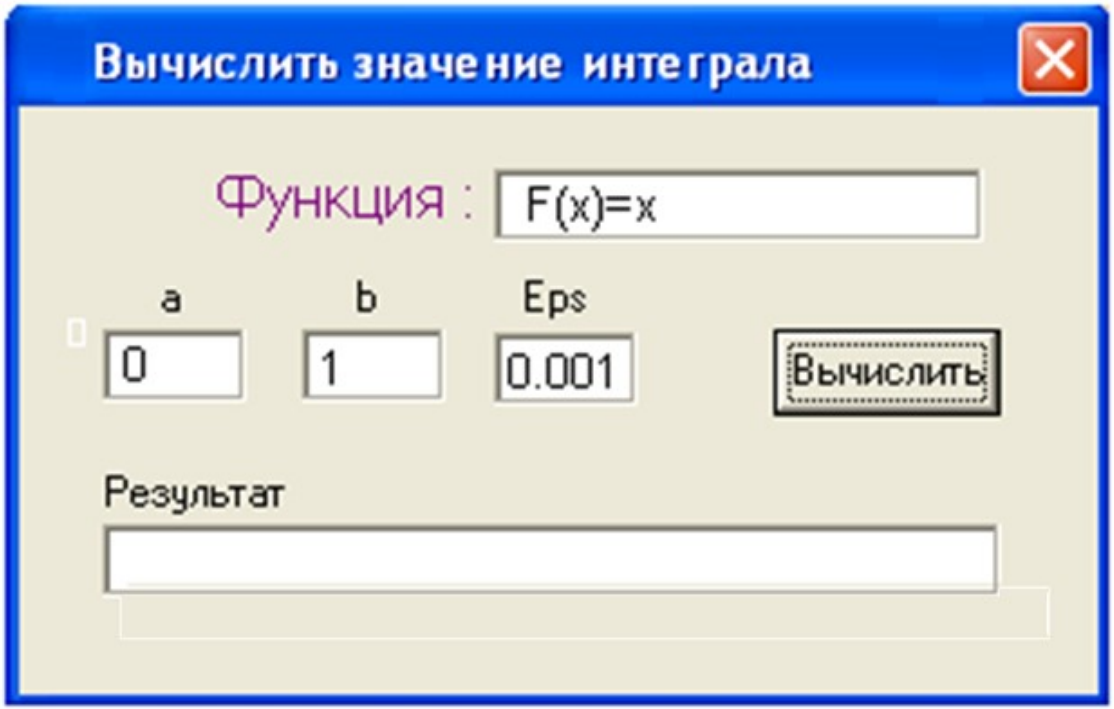
Опытный программист может найти ошибки "без всяких методов" (он подсознательно использует метод "предположение об ошибке").

Основа - интуиция.

.

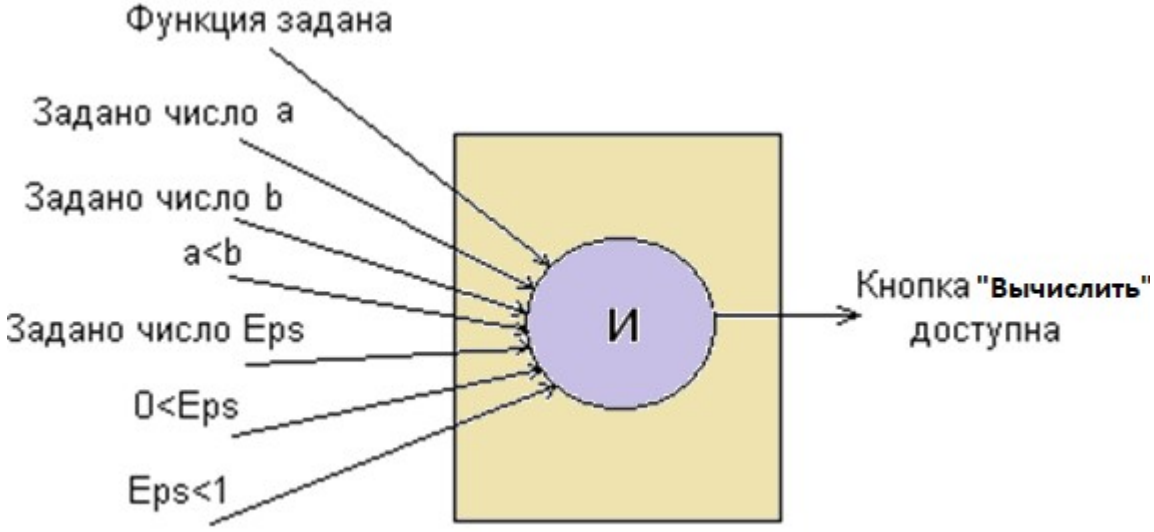
Рассмотрим пример, в котором программа должна вычислять с заданной точностью значение интеграла функции $F(x)=x$ на интервале от a до b.

На рисунке представлена форма интерфейса программы, из которой видно, что пользователь может задавать обычные и управляющие данные.



В начальном состоянии кнопка «Вычислить» недоступна и только в случае успешной проверки исходных данных, программа делает ее доступной для пользователя.

Рассмотрим причины, из-за которых кнопка «Вычислить» может быть недоступной. Из рисунка видно, что только при совпадении всех условий кнопка станет доступной и пользователь может получить результаты вычисления.



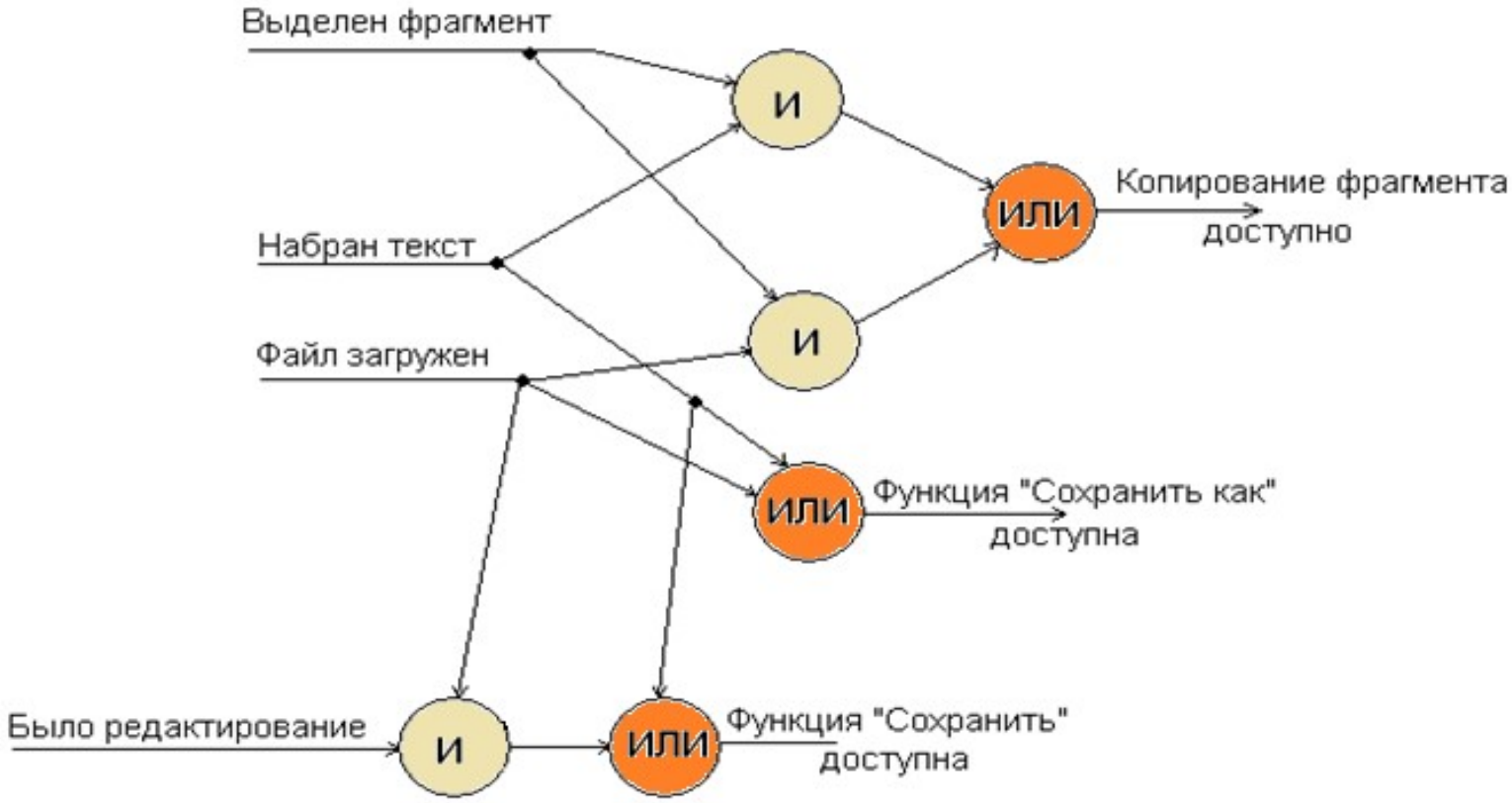
Причины доступности кнопки «Вычислить»

Есть еще причины, из-за которых, например, невозможно вычислить значение интеграла.

В частности, если на заданном интервале функция прерывается или заданную точность невозможно достигнуть на основе метода, который реализован в данной программе.

Рассмотрим пример, где осуществляется тестирование на управляющих данных, в частности, проверяется доступность функций управления редактированием.

На рисунке представлена логическая схема, на основе которой можно провести тестирование меню управления редактированием.



Логическая схема управления редактированием.

Вопросы для самопроверки

1. Перечислите методы ручного тестирования.
2. Какими возможностями обладают методы ручного тестирования?
3. Перечислите методы «белого ящика».
4. Какие методы «белого ящика» дают наибольшее количество тестов?
5. Приведите примеры перекрытий методов «белого ящика».
6. Какие методы «черного ящика» существуют?
7. Дайте анализ методов «черного ящика» применительно к различным видам данных.
8. Приведите пример возможного плана тестирования применительно к большим программным системам.
9. Какие существуют общие принципы тестирования программ?
10. Какиесуществуют виды ошибок?