

Функциональное тестирование

При **функциональном** тестировании логика программы не учитывается, а обращается внимание лишь на входные и выходные данные.

К методам функционального тестирования относится:

- 1) метод эквивалентного разбиения
- 2) анализ граничных значений

В **первом** методе область всех возможных наборов входных данных программы по каждому параметру разбивают на конечное число групп *классов эквивалентности (1-й этап)*, а затем на основе этих классов строятся сами тесты (*2-й этап*). Для разбиения на классы эквивалентности необходимо проанализировать входные условия и разбить их на два или более класса. Для любого условия существует правильный класс эквивалентности (содержащий корректные входные данные), и неправильный (содержащий ошибочные значения).

Рассмотрим классы эквивалентности для программы вычисления

$$y = 1/a + \sqrt{25-b*b}$$

Входное условие | Классы эквивалентности

|-----

| правильные | неправильные

1) a не должно быть равным 0 | $a < 0$ | $a = 0$

2) b должно находиться | |

в диапазоне от -5 до 5 | $-5 \leq b \leq 5$ | $b < -5$, $b > 5$

При построении тестов необходимо, чтобы каждый неправильный класс эквивалентности участвовал в тесте хотя бы один раз.

Таким образом, тесты могут быть следующими:

1) $a = 0$ $b = 4$

2) $a = 1$ $b = -7$

3) $a = 0$ $b = 6$

Во **втором** методе (анализ граничных значений) анализируются ситуации, возникающие на границах или вблизи границ классов эквивалентности.

Например, область корректных значений для переменной b - от -5 до 5 .

Следовательно, необходимы тесты, в которых b примет значения -5 , 5 , -5.01 и 5.01 .

Напомню, тестирование - это процесс обнаружения ошибки - первый этап отладки. Однако даже эффективное тестирование хотя и обнаруживает ошибки, но не может гарантировать их отсутствие.

Как уже упоминалось при тестировании модулей программного обеспечения, так же, как при проектировании и кодировании **возможно применение как восходящего, так и нисходящего подходов.**