

4.2. ホワイトボックステスト

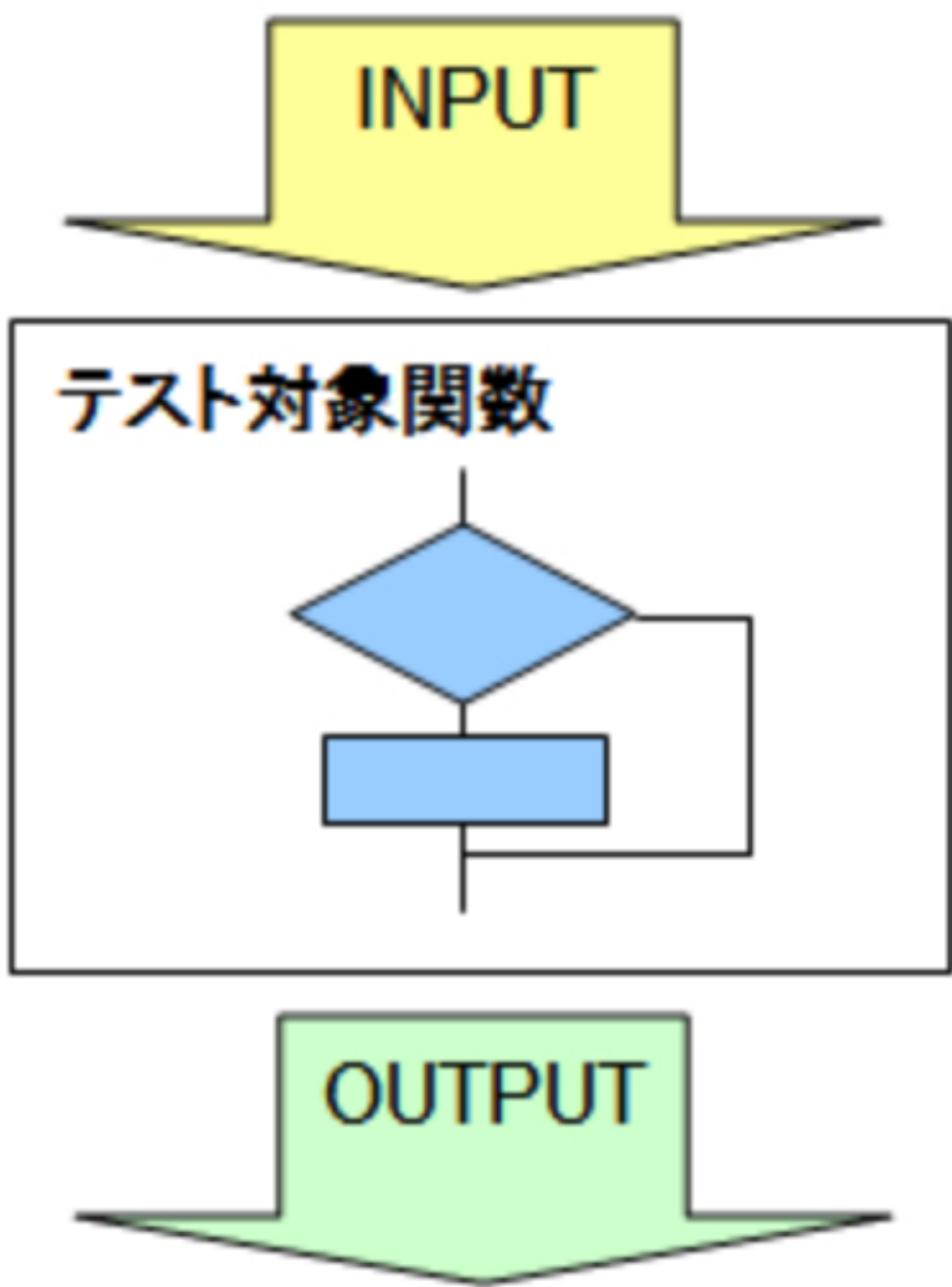
ブラックボックステストに対し、**ホワイトボックステスト**という観点があります。
ホワイトボックステストでは、テスト対象がどのような処理を行うのか理解した上で、そのとおりに動作していることを確認します。
つまり、テスト対象の処理が期待通りに実装されているのか確認する観点になります。

ホワイトボックステストとは

システムの内部の構造を理解した上で、それら一つ一つが意図したとおりに動作しているかを確認するテスト方法

ホワイトボックステスト

テスト項目作成の インプット情報	・ ユニットテストの場合はソースコード
長所	・ ソースコードを見ているため、ゼロ割になる箇所がないかなど実装観点のテストが可能
短所	・ 設計書を見ていないため、設計書通りに作成されているかのテストができない



ここで、「なんだか変だぞ?」と思われる方がいらっしゃるかもしれません。
「ソースコードは記述された処理どおりに動くんだから、期待通りに動作するのは当たり前じゃないか!」
はい。そのとおりです。
しかしホワイトボックステストでは、あえてソースコードが記述されたとおりに動作するかを確認します。

これは、以下の観点を確実にすることを目的としています。

- ・ 実行されていないコードがないかを確実にする
- ・ 危険コード（ゼロ割、オーバーフロー／アンダーフロー、メモリ破壊等）がないかをテストする

関数単位のユニットテストにおけるホワイトボックステストの場合は、以下のように定義できます。

ユニットテストにおけるホワイトボックステスト

ソースコードが期待した処理を行うように実装されていることを確認する。

Keyword

ホワイトボックステスト	システム内部の構造を理解した上で、それら一つ一つが意図したとおりに動作しているかを確認するテスト方法
-------------	--

補足

ISTQBでは、テスト技法を「経験ベース」「仕様ベース」「構造ベース」の3つに分類しています。
「構造ベース」のテスト技法はホワイトボックステスト技法にあたります。