

品質向上の  
トータルサポート企業  
バルテスがおくるミニ冊子  
**第4弾!!**

# 境界値分析・同値クラス分割の 基本の「き」

バルテス株式会社

## 目次

<b>1</b>	はじめに	1
<b>2</b>	境界値分析とは	2
<b>3</b>	境界値の注意点	4
<b>4</b>	境界値分析のメリット	5
<b>5</b>	同値クラス分割とは	6
<b>6</b>	同値クラス分割の具体例	8
<b>7</b>	境界値分析・同値クラス分割の具体例	10
<b>8</b>	まとめ	12
<b>9</b>	おわりに	13



# 1. はじめに

この冊子では、「**境界値分析**」「**同値クラス分割**」というテスト技法を紹介します。

一般的に、欠陥は条件の境目付近に潜んでいる可能性が高いと言えます。仕様書で「以下」と「未満」を取り違えてしまったり、コーディングで「 $\geq$ 」と「 $>$ 」の記述ミスをしったりといった**間違いが起こりやすい場所**だからです。

そこで、境界値分析を行うことによって、**欠陥が潜んでいるところを効率的にテスト**することができます。

また、ソフトウェアでは入力値や条件によって処理方法が決まりますが、その値の数や組み合わせは膨大なものになってしまいます。そのため、すべての入力値・条件を1つずつテストすることはできません。このような場合に、境界値分析・同値クラス分割を用いることで、**テストすべき条件や値を大幅に減らす**ことができます。

バルテスではソフトウェアテストの専門会社として、**テスト設計やテストケース作成時に境界値分析・同値クラス分割を活用**しています。本書ではソフトウェアテストに役立つツールとして、境界値分析・同値クラス分割の基本の「き」をお話していきます。

## ● 境界値とは・・・

プログラムへの入力に対する出力結果に、  
変化が生じる境界の値のこと

## ● 同値クラスとは・・・

何かの値を入力する際、  
出力結果が同じ結果となる値の集合

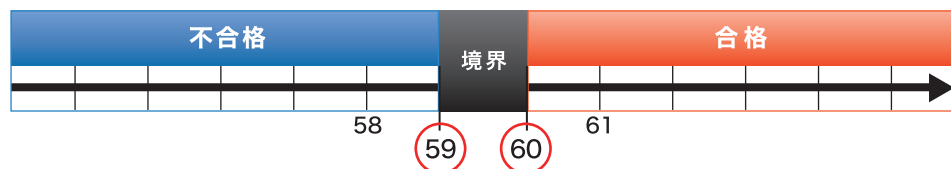


## 2.境界値分析とは

### ○ 境界値とは

ソフトウェアテストにおける境界値とは、「**プログラムの入力に対する出力結果に、変化が生じる境界の値**」を言います。

例えば、試験の点数を入力すると、合格か不合格かを判断するプログラムがあるとします。



上図で説明すると「59点」を入力したときの出力結果は不合格、「60点」を入力したときの出力結果は合格です。よって「59点」と「60点」で出力結果に変化が生じています。

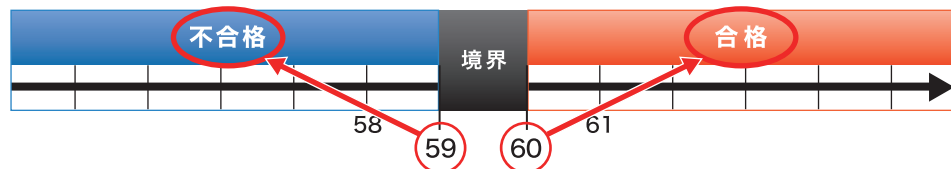
この「59点」と「60点」が「境界値」です。

### ○ 境界値分析とは

境界値分析とは、「**期待される出力結果に変化が生じる境界値を分析・テストすること**」を言います。

上記の合格・不合格判定プログラムを再び考えてみましょう。

例えば「59点」と入力したとき、出力結果が「不合格」となるか、また同様に「60点」と入力したとき、出力結果が「合格」となるかをテストします。



開発担当者もテスト担当者も無意識で実施していることが多いですが、改めてテスト技法と定義した上で、効果的に境界値分析を実施していきましょう。

それでは、境界値分析について、より詳しく見ていきます。

下記のような条件式があるとします。

$0 \leq x < 100$  ならば OK  
それ以外は NG (ただし、 $x$ は整数)

これを数直線に書き表すと下図になります。



### 1. 条件式「 $0 \leq x < 100$ 」の「 $0 \leq x$ 」の部分に注目

ここで結果が「OK」となる $x$ の最小値は「 $0 \leq x$ 」なので、「0」です。

また、「NG」となる最大値は「-1」なので、境界値はOKの「0」とNGの「-1」です。

### 2. 条件式「 $0 \leq x < 100$ 」の「 $x < 100$ 」の部分に注目

ここで結果が「OK」となる $x$ の最大値は「 $x < 100$ 」なので、「99」です。

また、「NG」となる最小値は「100」なので、境界値はOKの「99」とNGの「100」です。

結果、条件式「 $0 \leq x < 100$ 」における境界値は、出力結果OKとなる最小値「0」と最大値「99」、その境界に隣接し、出力結果がNGとなる「-1」と「100」になります。

それぞれの境界値を実際のプログラムに入力し、期待する出力結果通りになるかをテストします。



### 3.境界値の注意点

#### ● 認識の不一致が起こりやすい

➡ 等号表記や数直線を活用して、認識ずれを防ぎましょう。

境界を表す仕様の記述方法は「以上」、「以下」、「未満」、「より小さい」、「～を超える」、「～まで」などさまざまなものがありますが、こうした表現は認識にずれが生じやすくなります。

例えば、とある植物園の入場料の仕様では「13歳から20歳までの入場料は500円である」と記述されているとします。この場合13歳、20歳は500円で入園できるのかが曖昧です。



認識の不一致が発生しやすいのであれば、**できるだけ上流工程の段階で認識がずれない仕様を書く**べきです。以下にポイントをまとめます。

- ① 「<」「≤」などの記号を使用し、可能ならば「数直線」を添えること。  
その場合、端点が含まれるのか、含まれないのかが一目でわかるように「<」であれば「○」、「≤」であれば「●」を記入しておく。
- ② 整数なのか、小数点は何ケタまで有効なのか、分や秒まで含めるのか、といった「最小の単位」に気を付ける。数直線を書くときも同様である。

上記の植物園の例を数直線にしてみましょう。



有効入力値x  
 $13 \leq x < 20$   
 無効入力値x  
 $x < 13, 20 \leq x$   
 (ただし、xは整数)

13歳、20歳がどちらに含まれるのかが分かりますね。この場合、仕様作成者は「13歳は含む、20歳は含まない」という考えで記述していたようです。

数直線を使用するだけでも認識のずれは少なくなります。境界値が存在する仕様では手書きのメモでもいいので、数直線で示すようにしましょう。



## 4. 境界値分析のメリット

### ●境界値は不具合が発生しやすい部分である

➡ 最小の工数で効率よく不具合を発見できます。

境界値は、不等号の向きや範囲の設定ミスなど、不具合が多く発生しやすい部分です。

コーディングの際、境界は「=、≠、<、>、≤、≥」といった等号・不等号を用いて記述されます。仕様自体は正確に理解していても、タイプミスやコピー＆ペーストのミスによって、開発仕様書に記述されている条件とは異なるコーディングをしてしまうことがあります。

例えば、パスワードの文字数が「4文字以上、15文字以下 ( $4 \leq x \leq 15$ )」とされているとき、以下のようにコーディングされてしまう可能性があります。

if 文字数  $\geq 4$  and 文字数  $\leq 15$  then と書くはずが…

if 文字数  $\geq 4$  and 文字数  $\leq 16$  then

➡  $4 \leq x \leq 16$  に!

if 文字数  $\geq 4$  and 文字数  $< 15$  then

➡  $4 \leq x < 15$  に!

if 文字数  $\geq 4$  and 文字数  $= 15$  then

➡  $4 \leq x = 15$  に!

このように境界ではミスが起こりやすく、それが原因となって欠陥が作り込まれてしまいます。

よって、境界値分析を行い「境界」をテストすることで、最小の工数で効率よく不具合を発見することが可能となります。

また境界(値)があれば、**単体・統合・システム・受け入れ**といったどのテストレベルにおいてもテストが実施できます。

境界値分析は適用範囲が広いので、テストをする上で非常に有効な技法であると言えます。

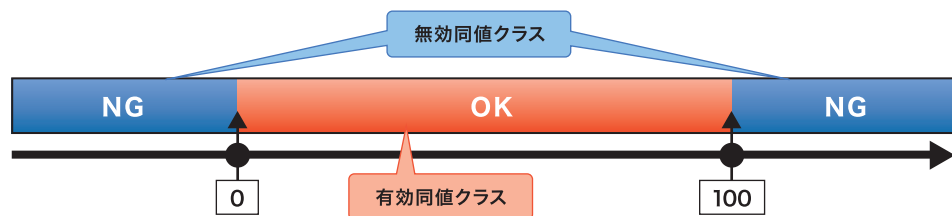


## 5.同値クラス分割とは

### ○ 同値クラス分割とは

p.1で、「出力結果が同じ結果となる値の集合」のことを同値クラスと呼ぶことを確認しました。  
同値クラス分割は、入力値による出力結果を、有効と無効の“同値クラス”に区分します。

入力値が $0 \leq x \leq 100$ のときOKと出力 … 有効  
入力値が $x < 0, 100 < x$ のときNGと出力 … 無効



上記の例で言えば、OKと出力される「 $0 \leq x \leq 100$ 」のグループが「有効同値クラス」、NGと出力される「 $x < 0, 100 < x$ 」のグループが「無効同値クラス」となります。

### ○ 同値クラス分割の特徴

同値クラスは、上図のように範囲が一続き(連続した値)になる場合もあれば、そうならない場合もあります。例えばパスワードに使用する文字で、l(英大文字のアイ) l(英小文字のエール) 1(数字のイチ)は「識別しにくいので無効である」と指定されている場合は、同値クラスは下記ようになります。

**有効同値クラス** … l以外の英大文字 / l以外の英小文字 / 1以外の数字  
**無効同値クラス** … 英大文字のl / 英小文字のl / 数字の1

このような分類も正しい同値クラスの分割です。 **範囲が一続きにならない場合があること**が同値クラスの大きな特徴であり、前述した境界値分析との異なる点でもあります。



## ○ 同値クラス分割テストとは

次に、同値クラス分割された値のテストとは、「**入力値に、各同値クラスのそれぞれの代表の値を入力し、出力結果を確認すること**」を言います。



上図では、入力値に有効同値クラス「0～100」の代表値を入力し、出力結果が正しいことを確認します。同様に、無効同値クラスである「有効同値クラス以外の値」を入力し、出力結果が正しいことを確認します。代表値は同値クラス分割の値の範囲内であれば、なんであってもかまいません（範囲の中間を選択する場合があります）。それは、何を入力するかではなく、**どの同値クラスの値を入力するか**ということが大切だからです。つまり、重要なことは、**同値クラス分割を漏れなく正確に行うこと**です。

## ○ メリット

本来であれば、入力可能なすべての入力値を入力し、出力結果を評価するのが理想です。ですが、それは現実的な話ではありません。そこで同値クラス分割を使用することによって、膨大な入力条件を有限個に絞り込み、**テスト工数を縮小**することができます。

## ○ 注意点

代表値として抽出した値はあくまで「代表」値です。つまり、**代表値をテストしたからといって、その同値クラスの入力値をすべてテストしたわけではない**ということです。

同値クラス分割は同じ結果になる値の集合としてテストしますが、1つ1つの値のテストは実施していないので、**同値クラスの中にイレギュラーな不具合があった場合、見逃してしまうリスクが潜んでいる**ことを忘れるべきではないでしょう。

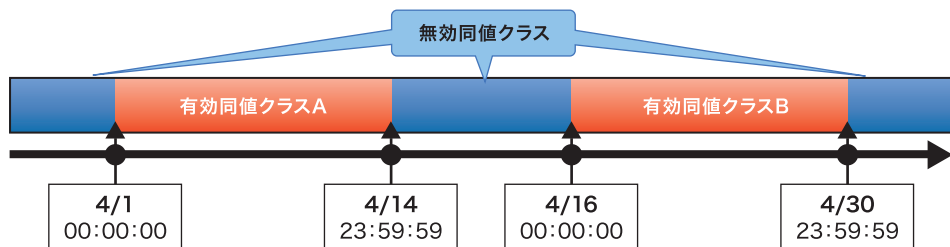


## 6.同値クラス分割の具体例

### 例1 インターネットによる商品販売システム

**第1期販売期間**    **開始:** 4月1日 0時00分00秒    **終了:** 4月14日 23時59分59秒

**第2期販売期間**    **開始:** 4月16日 0時00分00秒    **終了:** 4月30日 23時59分59秒  
(ただし、秒は整数とする)



#### 有効同値クラスA

4月1日0時00分00秒 ～ 4月14日23時59分59秒の代表値

#### 有効同値クラスB

4月16日0時00分00秒 ～ 4月30日23時59分59秒の代表値

#### 無効同値クラス

有効同値クラスA・有効同値クラスB以外の代表値

- 「開始:4月1日0時00分00秒 終了:4月14日23時59分59秒」の期間を **有効同値クラスA**として、この同値クラスの代表値を入力
- 「開始:4月16日0時00分00秒 終了:4月30日23時59分59秒」の期間を **有効同値クラスB**として、この同値クラスの代表値を入力
- それ以外を**無効同値クラス**とみなし、代表値を入力。この例では、3つの無効同値クラスが考えられるので、3つの同値クラスからそれぞれの代表値を入力

上記の3つのパターンの代表値を入力することで、同値クラス分割のテストを実施したことになります。

次に、数直線で表すことのできない(範囲が一続きにならない)場合の同値クラスの例を見てみましょう。

## 例2 パスワード入力システム

### 入力できる文字種

半角英数字(0～9、a～z、A～Z)のみ

入力文字数は4文字

この2つの条件を満たす有効同値クラスは「入力文字種が半角英数字(0～9、a～z、A～Z)かつ文字数が4文字の場合のみ」です。これに該当する代表値は「09aZ」「Az09」などが考えられます。

無効同値クラスには、有効同値クラス以外の値を入力するので、以下のようなものが考えられます。

### 有効同値クラス

- ①「文字数が4」かつ「半角英数字のみ」

### 無効同値クラス

- ①「文字数が0～3もしくは5以上」かつ「半角英数字のみ」
- ②「文字数が4」かつ「半角英数字以外の文字や記号」
- ③「文字数が0～3もしくは5以上」かつ「半角英数字以外の文字や記号」

以上の各クラスのパターンの代表値を抽出することで、同値クラス分割を行うことができます。

## ○実際の現場では

これまでの例の場合では、同値クラスを代表する入力値は、各クラスに対して1つでかまいませんでした。それは、きちんと同値クラス分割が行われていることが前提となっていたからです。

けれど、実際のテスト業務を行う際には、境界値分析テストなどと併用することが一般的です。そして境界値をテストすることは、同値クラスの代表値として境界値を選ぶ、ということでもあるのです。



## 7. 境界値分析・同値クラス分割の具体例

さて、本章では境界値分析・同値クラス分割を用いた具体例を確認します。

動物園のチケット販売機の料金計算です。仕様は以下の通りです。

チケット料金に関するテストを実施するために、境界値分析・同値クラス分割を用いて整理していきましょう。

### 動物園チケット料金計算仕様

#### チケット料金一覧

一般：2,000円  
 幼児：無料（6歳以下）  
 児童：半額（13歳未満）  
 学生：500円引（15歳以下）  
 未成年：300円引（20歳未満）  
 高齢者：半額（65歳以上）

ただし16時からは上記料金からさらに200円引

**開園時間** 9:30～17:00  
 （入園は16:30まで）

**販売時間** 9:00～16:30

・児童：7歳以上13歳未満と定義  
 ・学生：13歳以上15歳以下と定義

上記の仕様を整理するために、まずは「年齢」と「販売時間」のそれぞれについて、数直線を使って境界値を明らかにし、同値クラス分割を行いきましょう。

#### ① 年齢の境界値と同値クラス

0	6	7	12	13	15	16	19	20	64	65
無料		半額		500円引		300円引		一般料金		半額

#### ② 販売時間の境界値と同値クラス

0:00	8:59	9:00	15:59	16:00	16:30	16:31	23:59
購入不可		値引きなし		200円引		購入不可	

左記の①を横軸に、②を縦軸にまとめると、以下の<表A>となります。

表 A

年齢	0	6	7	12	13	15	16	19	20	64	65
割引 時刻	無料		半額		500円引		300円引		一般料金		半額
9:00 ～ 15:59	0円		1,000円		1,500円		1,700円		2,000円		1,000円
16:00 ～ 16:30	0円		800円		1,300円		1,500円		1,800円		800円

<表A>のうち、両端の境界値に注目し、さらにまとめ直したのが<表B>です。<表A>をより具体的に抽出した<表B>は、テストケースとしても活用することが可能です。

表 B

時刻 年齢	8:59	9:00	15:59	16:00	16:30	16:31
6	×	0	0	0	0	×
7	×	1,000	1,000	800	800	×
12	×	1,000	1,000	800	800	×
13	×	1,500	1,500	1,300	1,300	×
15	×	1,500	1,500	1,300	1,300	×
16	×	1,700	1,700	1,500	1,500	×
19	×	1,700	1,700	1,500	1,500	×
20	×	2,000	2,000	1,800	1,800	×
64	×	2,000	2,000	1,800	1,800	×
65	×	1,000	1,000	800	800	×

(×：購入不可)

以上のように、境界値分析・同値クラス分割は、「年齢・販売時間・割引」など、**複数の条件が重なった、複雑な仕様をまとめる際にも有効**です。

**誰が見ても一目で理解できる形で仕様を整理できる**のが、これらのテスト技法を用いる大きなメリットの1つです。



## 8. まとめ

境界値分析・同値クラス分割のポイントをおさらいしましょう。

### ●境界値分析とは

期待される出力結果に、変化が生じる境目を境界値と言い、その境界値を分析してテストすること。

- ➡ 境界値は不具合が発生しやすい部分です。
- ➡ 境界値分析を使用することにより、最小の工数で効率よく不具合を発見することが可能です。

### ●同値クラス分割とは

同じ出力結果となる入力値の集合を同値クラスと言い、その同値クラスを区分すること。

- ➡ 各同値クラスから代表する値を確認すれば、その同値クラス全体をテストしたことになる、という考え方です。これによりテスト工数を縮小できます。



## 9. おわりに

ここまで基本の「き」第4弾にお付き合いいただき、ありがとうございました。この冊子でご紹介させていただいたことは、境界値分析・同値クラス分割を使ったテスト設計のごく一部です。「隠れた境界値」など、紙幅の都合で割愛させていただいたトピックもあります。

バルテスではこのようなテスト技法を用いた効率的なテストを設計し、お客様に提供しています。

**また、バルテスでは様々なソフトウェアテストに関するセミナーも開催しています。**

今回の境界値分析・同値クラス分割も含めて、テスト技法に関しては、弊社セミナー「ソフトウェアテストの設計(上)」にて詳しく解説しています。機会があればぜひ受講をご検討ください。

**執筆協力**

石原一宏、江添智之、池邊歩美、泉谷有紀、富永恵里花

編集・発行 バルテス株式会社  
2014年（平成26年）6月25日 第1刷発行

連絡先 バルテス株式会社

**大阪本社**  
〒 550-0011 大阪市西区阿波座1-3-15JEL西本町ビル8F

**東京本社**  
〒 102-0083 東京都千代田区麹町1-10 麹町広洋ビル3F

TEL 06-6534-6561

FAX 06-6534-6562

ホームページ <https://www.valtes.co.jp>

バルテス

検索

## お問い合わせ先

### 電話番号

大阪

**06-6534-6562**

東京

**03-5210-2080**

### Eメールアドレス

**test-tech@valtes.co.jp**

本書は著作権法上の保護を受けています。  
本書の一部または全部について（ソフトウェアおよびプログラムを含む）、  
バルテス株式会社から文書による許諾を得ずに、いかなる方法においても  
無断で複写、複製することは禁じられています。



VALTES

Copyright © VALTES CO., LTD. All rights reserved.