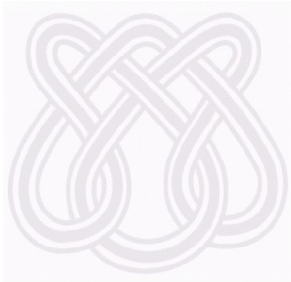


Exercício – Teste Funcional – Engenharia de Software II

Trabalho Desenvolvido pelos Alunos



*Ubiratan Soares (5634292)
Petrus Gomes (5634966)
Vinicius Grippa (5119050)
Lucas Licursi (5635101)
Diego Carvalho (5195672)*



12 de maio de 2009

O objetivo desse trabalho é exercitar o conceito de Teste Funcional de Software e obter casos de teste para a especificação de um programa através da técnica de Particionamento em Classes de Equivalência.

A partir da especificação do programa Cal fornecida em aula, pode-se notar três situações triviais para o exercício do programa Cal a partir do número de entradas, a saber:

- (1) Cal**
- (2) Cal *arg1***
- (3) Cal *arg1 arg2***

O número de argumentos tomados pelo programa Cal será adotado aqui como ponto de partida para identificar algumas classes de equivalência válidas e inválidas. Deve-se notar que a chamada de Cal com mais de três argumentos é irrelevante para esse estudo, uma vez que argumentos adicionais aos dois primeiros serão ignorados pelo programa, segundo a especificação.

Com base ainda nessa especificação, para um ou dois argumentos de entrada, identificamos as seguintes situações:

- ✓ **O argumento é não inteiro** : para (2) ou (3), ao menos um argumento não inteiro resultará em uma saída de erro;
- ✓ **O argumento é um ano** : para *arg1* em (2) ou *arg2* em (3), o argumento pode variar entre os valores de 1 a 9999;
- ✓ **O argumento é um mês** : para *arg1* em (3), o argumento pode variar entre os valores de 1 a 12;

Dessa maneira, as situações acima mostram que *arg1* ou *arg2* são entradas que, quando válidas, estabelecem um intervalo de valores válidos a serem aceitos, e assim sendo,

determinarão cada uma ao menos uma classe válida para os valores compreendidos nesses intervalos e duas classes inválidas para os valores acima e abaixo dos extremos do intervalo de aceitação. Essas conclusões são sintetizadas na tabela a seguir :

Args. de Cal	Classes Válidas	Classes Inválidas
-	Default (1)	Não existem
1	$1 \leq \text{arg1} \leq 9999$ (2)	$\text{arg1} < 1$ (3) $\text{arg1} > 9999$ (4) Não inteiro (5)
2	$1 \leq \text{arg2} \leq 12$ (6) $1 \leq \text{arg1} \leq 9999$ (7)	$\text{arg1} < 1$ (8) $\text{arg1} > 13$ (9) Não inteiro (10) $\text{arg2} < 1$ (11) $\text{arg2} > 9999$ (12) Não inteiro (13)

Tabela 1: Classes de Equivalências Básicas

Contudo, a especificação de Cal diz que as classes de equivalência (3), (6) e (7) não são tratadas da mesma maneira, devido a mudança de calendário e as novas regras impostas para o ano bissexto a partir de 1752. Dessa maneira, é necessário reparticionar essas classes em subclasses. Isso é feito através do particionamento do domínio de saída para cada caso. O refinamento das classes de equivalência para o caso em que Cal recebe um único argumento é colocado na tabela a seguir :

Valor de arg1	Saída Esperada
$1 \leq \text{arg1} \leq 1751$	Calendário Ano Normal se $\text{arg1} \bmod 4 \neq 0$ (2.1)
	Calendário Ano Bissexto se $\text{arg1} \bmod 4 = 0$ (2.2)
$\text{arg1} = 1752$	Calendário Ano Bissexto com mês de setembro truncado no mês de setembro, nos dias 03 a 13 (2.3)
$1753 \leq \text{arg1} \leq 9999$	Calendário Ano Normal se $\text{arg1} \bmod 4 \neq 0$ (2.4)
	Calendário Ano Bissexto se $\text{arg1} \bmod 4 = 0$ AND $\text{arg1} \bmod 100 \neq 0$ AND $\text{arg1} \bmod 400 \neq 0$ (2.5)
	Calendário Ano Normal se $\text{arg1} \bmod 4 = 0$ AND $\text{arg1} \bmod 100 = 0$ AND $\text{arg1} \bmod 400 = 0$ (2.6)

Tabela 2: Refinamento das Classes Válidas para um argumento de Cal

Seria ainda possível fazer um novo refinamento para o caso em que Cal é chamado com dois argumentos. Contudo, é fácil perceber que as classes de equivalência derivadas dessa abordagem estão plenamente cobertas pelas classes de equivalência já obtidas no refinamento anterior caso o programa Cal esteja implementado de maneira "inteligente", ou seja, ao executarmos por exemplo Cal 2009 estamos cobrindo Cal 1 2009 automaticamente.

Como o objetivo do Teste Funcional é ter por fonte de informação somente a especificação do programa, essas subclasses estão devidamente exercitadas nos casos de teste para entradas válidas de dois argumentos com base nas classes derivadas para um argumento, embora a tabela de refinamento que derivaria esses casos não tenha sido elaborada.

Assim sendo, obtemos as seguintes classes de equivalência para as entradas do programa:

Args. de Cal	Classes Válidas	Classes Inválidas
-	Default (1)	Não existem
1	$1 \leq \text{arg1} \leq 1751$ e $\text{arg1} \bmod 4 \neq 0$ (2.1)	$\text{arg1} < 1$ (3)
	$1 \leq \text{arg1} \leq 1751$ e $\text{arg1} \bmod 4 = 0$ (2.2)	
	$\text{arg1} = 1752$ (2.3)	$\text{arg1} > 9999$ (4)
	$1753 \leq \text{arg1} \leq 9999$ (2.4) e $\text{arg1} \bmod 4 \neq 0$	Não inteiro (5)
	$1753 \leq \text{arg1} \leq 9999$ (2.5) e $\text{arg1} \bmod 4 = 0$ e $\text{arg1} \bmod 100 = 0$ e $\text{arg1} \bmod 400 \neq 0$	
	$1753 \leq \text{arg1} \leq 9999$ (2.6) e $\text{arg1} \bmod 4 = 0$ e $\text{arg1} \bmod 100 = 0$ e $\text{arg1} \bmod 400 = 0$	
2	$1 \leq \text{arg2} \leq 12$ (6)	$\text{arg1} < 1$ (8)
		$\text{arg1} > 13$ (9)
		Não inteiro (10)
	$1 \leq \text{arg1} \leq 9999$ (7)	$\text{arg2} < 1$ (11)
		$\text{arg2} > 9999$ (12)
		Não inteiro (13)

Tabela 3: Domínio de Entrada de Cal particionado em Classes de Equivalência

A partir da tabela 3 ,buscamos derivar os casos de teste, de modo que:

- Os casos derivados estão preferencialmente o menor número de modo a cobrir todas as classes válidas;

- Para cada classe inválida, é desenhado um caso de teste;

Os casos de teste gerados são listados na tabela a seguir:

Número Caso	Número de Argumentos	Valor de Entrada	Classes Cobertas	Saída Esperada
1	0	-	1	Mês e Ano Corrente
2	1	<i>abcde</i>	5	ERRO : Argumento deve ser inteiro
3	1	<i>0</i>	3	ERRO : Argumento inválido
4	1	<i>10000</i>	4	ERRO : Argumento inválido
5	1	<i>1579</i>	2.1	Calendário 1579 (Não bissexto)
6	1	<i>1700</i>	2.2	Calendário 1700 (bissexto)
7	1	<i>1752</i>	2.3	Calendário 1752 - setembro truncado (bissexto)
8	1	<i>1995</i>	2.4	Calendário 1995 (não bissexto)
9	1	<i>2004</i>	2.5	Calendário 2004 (bissexto)
10	1	<i>2000</i>	2.6	Calendário 2000 (não bissexto)
11	2	<i>abc</i> <i>1200</i>	10 2.1	ERRO : Argumento 1 inválido
12	2	<i>1565</i> <i>cdf</i>	2.1 13	ERRO : Argumento 2 inválido
13	2	<i>fgh</i> <i>ijk</i>	10 13	ERRO : Argumentos inválidos
14	2	<i>0</i> <i>1800</i>	8 7	ERRO : Argumento 1 inválido
15	2	<i>13</i> <i>1925</i>	9 7	ERRO : Argumento 1 inválido
16	2	<i>1</i> <i>0</i>	6 11	ERRO : Argumento 2 inválido
17	2	<i>10</i> <i>12451</i>	6 12	ERRO : Argumento 2 inválido
18	2	<i>5</i> <i>2009</i>	2.4 6	Calendário de maio de 2009 (não bissexto)
19	2	<i>2</i> <i>1998</i>	2.5 6	Calendário de fevereiro de 1998 (bissexto)
20	2	<i>2</i> <i>2000</i>	2.6 6	Calendário de fevereiro de 2000 (não bissexto)
21	2	<i>7</i> <i>1557</i>	2.1 6	Calendário de julho de 1557 (não bissexto)
22	2	<i>8</i> <i>1400</i>	2.2 6	Calendário de agosto de 1400 (bissexto)
23	2	<i>9</i> <i>1752</i>	2.3 6	Calendário truncado de setembro de 1752 (bissexto)

Tabela 4: Casos de Teste para o Programa Cal