Actividades UT6

# Caja Negra

## Resumen de la teoría de caja negra

**Determinar las clases equivalencia:**

* Rango: una válida y dos no válidas.
* Valor específico: una válida y una no válida.
* Conjunto de valores:
  + Si cada elemento se trata igual,una válida y otra no válida.
  + Si cada elemento se trata diferente, una válida por cada elemento y otra no válida
* Lógica: una válida y otra no válida.

**Diseñar los casos de prueba**

* **Para las clases válidas:** diseñar el menor número de casos posibles. En cada caso de prueba intentarmos abarcar el mayor número de cláses válidas.
* **Para las clases no válidas:** diseñaremos un caso de prueba por cada clase no válida, introduciendo el valor de la clase no válida a probar y en las demás entradas valores correspondientes a cláses válidas.

## Ejercicio 1 - Aplicación bancaria

Datos de entrada:

* Código de área: número de 3 dígitos que no empieza por 0 ni por 1
* Nombre de identificación de operación: 6 caracteres
* Órdenes posibles: “cheque”, “depósito”, “pago factura”, “retirada de fondos”

Diseño de clases de equivalencias

Rango:

* 1 clase válida: 200 < código < 999
* 2 clases no válidas: código < 200; código > 999

Lógica (Formato datos):

* 1 clase válida: número
* 2 clase no válida: no es número

Nombre de identificación:

Valor específico:

* 1 clase válida: 6 caracteres
* ½ clases no válidas: más de 6 caracteres; menos de 6 caracteres

Órdenes posibles:

Conjunto de valores:

* 4 clases válidas: 4 órdenes válidas
* 1 clase no válida: orden no válida

|  |  |  |
| --- | --- | --- |
| Datos de entrada | Clases válidas | Clases no válidas |
| Código de area | (1)200 <=código <=999 | (2)código < 200  (3)código > 999  (4)no es numérico |
| Identificación | (5)6 caracteres | (6)menos de 6 caracteres  (7)más de 6 caracteres |
| Orden | (8)”cheque”  (9)”depósito”  (10)”pago factura”  (11)”retirada de fondos” | (12)ninguna orden válida |

CASOS DE PRUEBA VÁLIDAS

|  |  |  |  |
| --- | --- | --- | --- |
| Código | Identificación | Orden | Clases cubiertas |
| 300 | AAAAAA | “Depósito” | 1. (5) (9) |
| 400 | Viajes | “Cheque” | 1. (5) (8) |
| 500 | Coches | “Pago de factura” | 1. (5) (10) |
| 600 | Comida | “Retirada de fondos” | 1. (5) (11) |

CASOS DE PRUEBA NO VÁLIDAS

|  |  |  |  |
| --- | --- | --- | --- |
| Código | Identificación | Orden | Clases cubiertas |
| 180 | Viajes | “Pago factura” | (2)(5)(10) |
| 1032 | Nómina | “Depósito” | (3)(5)(9) |
| XY | Compra | “Retirada fondos” | (4)(5)(11) |
| 350 | A | “Depósito” | (1)(6)(9) |
| 450 | Regalos | “Cheque” | (1)(7)(8) |
| 550 | Casas | &%4 | (1)(5)(12) |

## Ejercicio 2 – Aplicación para sorteo de vivienda

El programa debe guardar las puntuaciones en el baremo de una serie de solicitantes de vivienda, y nos debe decir si han resultado beneficiados o no con la adjudicación de una vivienda subvencionada por el Ayuntamiento (La adjudicación se materializa para puntuaciones en el baremo iguales o superiores a 5).

El programa admite como entrada el DNI y un número entre 0 y 10 sin decimales.

CLASES DE EQUIVALENCIA

DNI

Longitud:

* 2 clases válidas: una con 8 caracteres y otra con 9
* 2 clases no válidas: una con 7 o menos caracteres y otra con 10 o más

Lógica (Formato de datos):

* 2 clases válidas: una con 7 cifras y una letra y otra con 8 y la letra, coinciden con las anteriores por lo que no hace falta repetirlas
* 2 clases no válidas: cualquier combinación diferente de 8 y 9 cifras.

BAREMO

Rango:

* Válida: con un valor entre 0 y 10
* No válidas: una con valor negativo y otra con un valor superior a 10

Rango (con decimales):

* No válida: un número entre 0 y 10 con decimales.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Condición | Clases válidas | Número | Clases no válidas | Número |
| DNI | 7 cifras más una letra  8 cifras más una letra | 1  2 | Menos de 8 caracteres  Más de 9 caracteres  8 caracteres que no sean 7 cifras más letra  9 caracteres que no sean 8 cifras más letra | 3  4  5  6 |
| Puntuación | Un valor x con rango 0<=x<=5  Un valor x con rango 5<=x<=10 | 7  8 | Un valor <0  Un valor >10  Un valor con decimales | 9  10  11 |

|  |  |  |
| --- | --- | --- |
| Casos de pruebas válidos | Clases de equivalencia cubiertas | Resultado |
| 1234567 A 4 | 1 7 | No adjudicado |
| 12345678 A 9 | 2 8 | Adjudicado |

|  |  |  |
| --- | --- | --- |
| Casos de prueba no válidos | Clases de equivalencia cubiertas | Resultado |
| 123456 A 2 | 3 7 | Dato no válido |
| 1234567890 A 2 | 4 70 | Dato no válido |
| 1234abc5 2 | 5 7 | Dato no válido |
| Abcd12345 6 | 6 8 | Dato no válido |
| 1234567ª -5 | 1 9 | Dato no válido |
| 12345676ª 20 | 1 10 | Dato no válido |
| 1234567ª 7,5 | 1 11 | Dato no válido |

Versión (obligar meter al 0)

|  |  |  |
| --- | --- | --- |
| Condición | Clases válidas | Clases no válidas |
| DNI | 8 cifras más una letra(1) | <9 caracteres (2)  >9 caracteres (3)  9 caracteres que no sean 8 números más una letra (4) |
| Puntuación Baremo | Un valor x con rango 0<=x<=5  (5)  Un valor x con rango 5<=x<=10 (6) | Un valor <0 (7)  Un valor >10 (8)  Un valor con decimales (9) |

|  |  |  |
| --- | --- | --- |
| Casos de prueba válidos | Clases probadas | Resultado esperado |
| 12345678 A 4 | 1 y 5 | Vivienda no adjudicada |
| 12345678 A 7 | 1 y 6 | Vivienda adjudicada |

|  |  |  |
| --- | --- | --- |
| Casos de prueba no válidos | Clases probadas | Resultado esperado |
| 1234567 4 | 2 y 5 | Error de formato DNI |
| 0123456789 4 | 3 y 5 | Error de formato DNI |
| 012345678 4 | 4 y 5 | Error de formato DNI |
| 012345678 -1 A | 1 y 7 | Error en la puntuación |
| 01234567 A 11 | 1 y 8 | Error en la puntuación |
| 01234567 A 5.5 | 1 y 9 | Error en la puntuación |

## Ejercicio 3 – Inversión de un número

Un programa recibe como entrada un número entero y positivo de mínimo 2 cifras y de máximo 9 cifras y devuelve el número resultante de invertir sus cifras.

Si no se introduce un valor acorde a lo descrito rango, etc.), el módulo devolverá el valor “error”. No olvidar el análisis de valores límite (por ejemplo: flotantes y/o caracteres, valores fuera de rango)

ENTRADA

Rango:

* Válida: número entre 10 y 999999999
* No válida: número menor de 10 o mayor de 999999999.

Lógica:

* Válida: combinación entre 2 y 9 caracteres formado por números
* No válida: combinación entre 2 y 9 caracteres formado por números y letras, o formado por caracteres que no sean números

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Condición | Clase válida | Número | Clase no válida | Número |
| Rango | Número > 10  Número < 999999999 | 1  2 | Número < 10  Número > 999999999 | 3  4 |
| Lógica | Combinación entre 2 y 9 caracteres formado por números | 5 | Combinación entre 2 y 9 caracteres formado por números y letras  Combinación entre 2 y 9 caracteres formado por letras | 6  7 |

|  |  |  |
| --- | --- | --- |
| Casos de prueba válidos | Clases probadas | Resultado esperado |
| 12345 | 1 y 5 | Correcto |
| 5369845 | 2 y 5 | Correcto |

|  |  |  |
| --- | --- | --- |
| Casos de prueba no válidos | Clases probadas | Resultado esperado |
| 156780987654 | 1 y 4 | Error |
| 12342SA | 1, 2, 6 | Error |
| 2 | 3 | Error |
| AHJSGDJS | 5 y 7 | Error |

## Ejercicio 4 – Hora

Tenemos un pequeño módulo que lee una hora en formato de hh:mm e indica si la hora es correcta.

Se anexa a continuación la pantalla:

Aplicación

Descripción generada automáticamente con confianza media

Si no se introduce un valor acorde a lo descrito (por ejemplo: flotantes y/o caracteres, valores fuera de rango, etc.), el módulo devolverá el valor “error”. Genere la tabla de clases de equivalencia y los casos de prueba (no olvide el análisis de valores límite).

Solución:

n = número de parámetros

A = conjunto de valores permitidos: los números naturales más el cero. A = N U {0}.

ENTRADA

Rango:

* Válida: dos números (el primero entre 00 y 23, y el segundo entre 00 y 59).
* No válida: dos números (el primero menor de 00 o mayor de 23, y el segundo menor de 00 y mayor de 59).

Lógica:

* Válida: conjunto de 2 caracteres formado únicamente por números.
* No válido: combinación de caracteres formados por letras y números, o formado por letras.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Condición | Clase válida | Número | Clase no válida | Número |
| Rango | Primer número >= 00  Primer número <= 23  Segundo número >= 00  Segundo número <= 59 | 1  2  3  4 | Primer número <00  Primer número > 24  Segundo número < 00  Segundo número > 59 | 5  6  7  8 |
| Lógica | Conjunto de caracteres formados por dos números | 9 | Conjunto de caracteres formado por letras y números  Conjunto de caracteres formado por letras | 10  11 |

|  |  |  |
| --- | --- | --- |
| Clases válidas | Clases probadas | Resultado esperado |
| 01:32 | 1, 3 y 9 | Correcto |
| 10: 54 | 2, 3 y 9 | Correcto |
| 12: 32 | 1, 4 y 9 | Correcto |
| 20:59 | 2, 4 y 9 | Correcto |

|  |  |  |
| --- | --- | --- |
| Clases no válidas | Clases probadas | Resultado esperado |
| -12:34 | 2, 5 y 9 | Error |
| 25:35 | 2, 6 y 9 | Error |
| 12: -45 | 1, 7 y 9 | Error |
| 12: 98 | 1, 8 y 9 | Error |
| A1: 43 | 3, 10 | Error |
| A | 11 | Error |