novática
Revista de la Asociación de Técnicos de Informática

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software).

<http://www.ati.es/novatica/>
<http://www.ati.es/reicis/>

ATI es miembro fundador de **CEPIS** (*Council of European Professional Informatics Societies*) y es representante de España en **IFIP** (*International Federation for Information Processing*); tiene un acuerdo de colaboración con **ACM** (*Association for Computing Machinery*), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AI2**, **ASTIC**, **RITSI** e **Hispalinux**, junto a la que participa en **ProInnova**.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
Plaza de España 6, 2ª planta, 28008 Madrid
Tlfn.914029391; fax.913093685 <novatica@ati.es>
Composición, Edición y Redacción ATI Valencia
Av. del Reino de Valencia 23, 46005 Valencia
Tlfn. 963740173 <novatica_prod@ati.es>
Administración y Redacción ATI Cataluña
Calle Ávila 50, 3a planta, local 9, 08005 Barcelona
Tlfn.934125235; fax 934127713 <secregen@ati.es>
Redacción ATI Andalucía <secreand@ati.es>
Redacción ATI Galicia <secregal@ati.es>
Suscripción y Ventas <novatica.subscripciones@atinet.es>
Publicidad Plaza de España 6, 2ª planta, 28008 Madrid
Tlnf.914029391; fax.913093685 <novatica@ati.es>
Imprenta: Derra S.A., Juan de Austria 66, 08005 Barcelona.
Depósito legal: B 15.154-1975 – ISSN: 0211-2124; CODEN NOVAEC
Portada: "Sueños probatorios" - Concha Arias Pérez / © ATI
Diseño: Fernando Agresta / © ATI 2003

sumario

Nº 224, julio-agosto 2013, año XXXIX

Tema del próximo número: "Empresa 2.0"

El problema del CUIT

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas

Laboratorio de Investigación de Software MsLabs, Dpto. Ing. en Sistemas de Información, Facultad Regional Córdoba - Universidad Tecnológica Nacional (Argentina)

<jotacastillo@gmail.com>,
<diegojserrano@gmail.com>,
<ing.marinacardenas@gmail.com>

El enunciado de este problema apareció en el número 223 de ***Novática*** (mayo-junio 2013, p.82). Previamente, fue publicada otra versión del mismo enunciado conteniendo una errata relevante en Novática nº 222, por lo que remitimos a todos los lectores al enunciado contenido en dicho número 223.

El CUIT (clave única de identificación tributaria) es una clave numérica que se utiliza en Argentina para identificar unívocamente tanto a personas físicas como jurídicas, y es el equivalente al NIF (número de identificación fiscal) empleado en España.

El problema planteado presenta como objetivo determinar si un número de CUIT es o no válido de acuerdo a las reglas de formación y validación de las claves CUIT.

Como información se conoce que las claves son números de 11 dígitos, de los cuales el último dígito se lo conoce como dígito verificador y es utilizado para realizar control de errores en el CUIT que permiten determinar si los 10 dígitos anteriores son o no válidos. Para resolver este problema es necesario implementar el algoritmo de validación provisto en el enunciado que consiste en multiplicar cada uno de los 10 dígitos de datos (se descarta el último dígito de verificación) por la secuencia 5432765432 e ir acumulando dicho producto. Nótese que la secuencia comienza en 5 y disminuye hasta 2 (inclusive), y luego reinicia en 7 y desciende nuevamente hasta 2 (inclusive). Esta observación nos permitirá emplear convenientemente una estructura repetitiva para el cálculo y acumulación de los resultados parciales.

Otra solución alternativa, sería utilizar un vector de coeficientes y colocar en la posición 0 el 5, en la posición 1 el 4, y así sucesivamente siguiendo la secuencia 5432765432.

Por ejemplo, `int []v_coeficientes=new int[] {5,4,3,2,7,6,5,4,3,2};` para utilizarla posteriormente dentro del segundo `for` anidado, lo cual permitiría ahorrarnos 3 líneas de código. Dejamos al lector el ensayo de esta implementación.

Una vez calculado el acumulador, se procede a calcular su resto módulo 11 el cual siempre dará un número entre 0 y 10, a este valor lo llamamos "resto" en el código de la solución. Seguidamente se debe calcular el dígito verificador como la diferencia entre el 11 y el resto, el cual dará como resultado un número entre 1 y 11, a este valor lo llamamos "verificador" en la solución propuesta.

Ya que el código de verificación debe ser solamente un dígito, el algoritmo establece que si el resultado obtenido en la verificación es 11, entonces el valor que se debe asignar al código verificador es 0, y finalmente, si el valor obtenido en la verificación es 10, entonces el valor que se debe asignar al código verificador es 9.

Estos resultados se pueden observar en aritmética modular, cuya aproximación moderna fue desarrollada por el célebre Carl F. Gauss en el año 1801, en su libro "*Investigaciones Aritméticas*".

En cuanto al diseño de la solución, y debido a la sencillez del problema, la misma hace uso solamente de una clase denominada CUIT, en la que su método principal se encarga de la entrada de datos, y en base a ellos, de la validación e impresión de los mensajes "*CORRECTO*" e "*INCORRECTO*".

A continuación se expone el código de la solución del problema en el lenguaje de programación Java.

```
import java.util.Scanner;

public class CUIT {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int C = sc.nextInt();
        int multiplicador = 5;
        int acumulador = 0;
        for (int i = 0; i < C; i++) {
            String cuit = sc.next();
            for (int j = 0; j < 10; j++) {
                char c = cuit.charAt(j);
                int digito = (int) c - '0';
                acumulador += digito * multiplicador;
                multiplicador--;
                if (multiplicador == 1) {
                    multiplicador = 7;
                }
            }
            int resto = acumulador % 11;
            int verificador = 11 - resto;
            if (verificador == 11) verificador = 0;
            if (verificador == 10) verificador = 9;
              int verificadorIngresado = (int)
cuit.charAt(10)-'0';
            if (verificador == verificadorIngresado)
{
                System.out.println("CORRECTO");
            } else {
                System.out.println("INCORRECTO");
            }
        }
    }
}
```