

36019475. CSIFC03. MP0485. Programación

**XUNTA DE GALICIA**CONSELLERÍA DE CULTURA, EDUCACIÓN
E ORDENACIÓN UNIVERSITARIA

Páxina principal ► Os meus cursos ► Formación Profesional a Distancia ► Curso 2016-2017 ►
36019475 IES de Rodeira ► CSIFC03 Desenvolvemento de aplicacións web ►
125_36019475_ZSIFC03_MP0485_A ► Unidade didáctica 5 ► Tarea para PROG05.

Tarea para PROG05.

Tarea para PROG05.

Detalles de la tarea de esta unidad.

Enunciado.

A lo largo de esta unidad has ido aprendiendo a crear tus propias clases así como sus distintos miembros (atributos y métodos). Has experimentando con la encapsulación y accesibilidad (modificadores de acceso a miembros), has creado miembros estáticos (de clase) y de instancia (de objeto), has escrito constructores para tus clases, has sobrecargado métodos y los has utilizado en pequeñas aplicaciones. También has tenido tu primer encuentro el concepto de herencia, que ya desarrollarás en unidades más avanzadas junto con otros conceptos avanzados de la Programación Orientada a Objetos.

Una vez finalizada la unidad se puede decir que tienes un dominio adecuado del lenguaje Java como para desarrollar tus propias clases y utilizarlas en una aplicación final que sea capaz de manipular un conjunto de datos simple. Dada esa premisa, esta tarea tendrá como objetivo escribir una pequeña aplicación en Java empleando algunos de los elementos que has aprendido a utilizar.

Se trata de desarrollar una aplicación Java en consola que permita gestionar una cuenta bancaria. Mediante un menú se podrán realizar determinadas operaciones:

1. **Ver el número de cuenta completo (CCC - Código Cuenta Cliente).**
2. **Ver el titular de la cuenta.**
3. **Ver el código de la entidad.**
4. **Ver el código de la oficina.**
5. **Ver el número de la cuenta** (solamente el número de cuenta, sin entidad, oficina ni dígitos de control).
6. **Ver los dígitos de control de la cuenta.**
7. **Realizar un ingreso.** Habrá que solicitar por teclado la cantidad que se desea ingresar.
8. **Retirar efectivo.** Habrá que solicitar por teclado la cantidad que se desea retirar.
9. **Consultar saldo.**
10. **Salir de la aplicación.**

Antes de que aparezca este menú, el programa tendrá que solicitar al usuario los siguientes datos:

- Nombre del **titular** de la cuenta (con un máximo de caracteres).
- **Código cuenta cliente (CCC)** de la cuenta completo (**entidad-oficina-dígitos de control-cuenta**).

El programa deberá asegurarse que el CCC es válido mediante la comprobación de:

- El **formato** (cuatro dígitos de entidad, cuatro dígitos de oficina, dos dígitos de control y diez dígitos de número de cuenta).
- Los **dígitos de control** son válidos.

Además del programa principal de la aplicación (clase con una función `main`), habrá que escribir una clase `CuentaBancaria` que proporcione todas las herramientas necesarias para trabajar con este tipo de información:

- Constructor (o constructores) adecuados.
- Almacenamiento del nombre del titular (atributos).
- Almacenamiento del código de cuenta (atributos).
- Almacenamiento del saldo actual (atributos).
- Gestión de ingresos y depósitos (métodos de interfaz pública).
- Obtención del saldo (métodos de interfaz pública).

- Aquellas herramientas auxiliares necesarias para poder trabajar cómodamente con el objeto. Algunas de esas herramientas podrán ser públicos y otras quizá no. Algunas podrán ser específicas de clase y otras podrán ser de objeto (métodos de objeto privados, métodos estáticos públicos, etc.).

Para trabajar con el número de cuenta debes utilizar el modelo de **Código Cuenta Cliente (CCC)**, que está formado por cuatro campos: entidad - sucursal - dígito de control - número de cuenta. La idea es que puedas introducir el código de cuenta completo y que la clase disponga de un mecanismo para comprobar que ese código es válido. Si el código no es válido, se debería generar una **excepción** (y por supuesto no almacenar ese código de cuenta). Para ello podrías tener, por ejemplo, un **método estático** que permita validar códigos de cuenta.

En general, deberías incluir **excepciones** para controlar aquellos casos en los que el uso de un método no sea posible (intentar sacar más dinero del que hay en el saldo, intentar introducir un titular con más caracteres de los permitidos, intentar ingresar o retirar una cantidad negativa, etc.).

El código fuente Java de esta clase debería incluir **comentarios** en cada atributo (o en cada conjunto de atributos) y método (o en cada conjunto de métodos del mismo tipo) indicando su utilidad. El programa principal también debería incluir algunos comentarios explicativos sobre su funcionamiento y la utilización de objetos de la clase

CuentaBancaria .

Además del programa deberás escribir también un informe con todas las consideraciones oportunas que se necesiten para entender cómo has realizado la tarea.

El proyecto deberá contener al menos dos archivos fuente Java:

- **Programa principal** (clase con método `main` : **AplicacionCuentaBancaria.java**).
- La clase **CuentaBancaria** (**CuentaBancaria.java**).

El documento que contendrá el informe lo elaborarás con un procesador de texto. Debe ser de tipo ".doc" (Microsoft Word) o de tipo ".odt" (OpenOffice.org). Debe tener tamaño de página A4, estilo de letra Times New Roman, tamaño 12 e interlineado normal.

Criterios de puntuación. Total 10 puntos.

Para poder empezar a aplicar estos criterios es necesario que la aplicación compile y se ejecute correctamente en un emulador. En caso contrario la puntuación será directamente de 0,00.

Criterios de puntuación.		
La clase CuentaBancaria	dispone de todos los atributos necesarios.	1,00
La clase CuentaBancaria	dispone de al menos un constructor y funciona correctamente.	1,00
La clase CuentaBancaria	dispone de los métodos públicos de interfaz necesarios y funcionan correctamente.	4,00
La clase CuentaBancaria	es capaz de validar un CCC .	2,00
Los métodos de la clase CuentaBancaria	son capaces de lanzar excepciones si se produce alguna situación anómala.	1,00
La clase CuentaBancaria	dispone de métodos estáticos públicos para proporcionar herramientas de gestión útiles al código de fuera de la clase.	1,00
No se han incluido comentarios en la clase CuentaBancaria tal y como se ha pedido en el enunciado.		-1,00
No se han incluido comentarios apropiados en el programa principal describiendo el funcionamiento de éste.		-1,00
No se ha entregado el informe explicativo o se trata de un informe explicativo insuficiente.		-2,00
El programa principal no es capaz de crear un objeto de la clase CuentaBancaria .		-5,00
Alguna de las opciones de menú pedidas en el enunciado (menú del programa principal) no funciona correctamente.		-1,00 por cada opción
Total		10,00

Dado que algunos criterios de puntuación son negativos, podría suceder que el balance final fuera negativo. En tal caso la puntuación final será simplemente de 0,00.

Recursos necesarios para realizar la Tarea.

- Ordenador personal.
- JDK y JRE de Java SE.
- Entorno de desarrollo NetBeans con las funcionalidades necesarias para desarrollar y emular midlets.

Consejos y recomendaciones.

- Básate en los diferentes ejemplos que has tenido que probar durante el estudio de la unidad. Algunos de ellos te podrán servir de mucha ayuda, así que aprovéchalos.
- El ejercicio resuelto de la clase **DNI**, en el cual se hacen comprobaciones de entrada, puede servirte de base para la comprobación de la validez de un **CCC**.
- Puedes obtener información acerca del funcionamiento de la **CCC** y de cómo calcular los dígitos de control del siguiente artículo de **Wikipedia**:
Wikipedia: Código Cuenta Cliente.
- Puedes generar cuentas bancarias válidas (o comprobarlas) para hacer pruebas en tu programa desde el siguiente enlace:
Generador/validador de cuentas bancarias.

Indicaciones de entrega.

Una vez que tengas terminados el programa (carpeta de proyecto Netbeans, incluyendo todos los archivos fuente y todos los recursos) y el documento explicativo, comprime ambos en un único archivo comprimido. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la quinta unidad del MP de PROG**, debería nombrar esta tarea como...

sanchez_manas_begona_PROG05_Tarea

Estado da entrega

Estado da entrega	Sen intentos
Estado das cualificacións	Sen cualificar
Data límite	Martes, 6 de Decembro do 2016, 20:55
Tempo restante	A tarefa venceu fai: 201 días 20 horas

Engadir entrega

Facerlle cambios á súa entrega

NAVEGACIÓN

- Páxina principal
- A miña área persoal
- Páxinas do sitio
- O meu perfil
- Curso actual
- 125_36019475_ZSIFC03_MP0485_A

Participantes

Distincións

Xeral

Unidade didáctica 1















Unidade didáctica 2

Unidade didáctica 3

Unidade didáctica 4

Unidade didáctica 5

Orientaciones para el alumnado. PROG05.

 PROG05 Guiada.- Desarrollo de clases. Actividades presenciales de la UD5 en la tutoría c... Otello Foro para PROG05. Mapa conceptual para PROG05. DAW05 Completa.- Desarrollo de clases. Recursos complementarios UD05. **Tarea para PROG05.** Solución Tarea Tarefa a Entregar Solución do Exercicio Examen de la UD05. Modelo de Exame Indicacións e Explicacións Á solución Ficheiros Auxiliares do Modelo de Exame Solucións ao Modelo de Exame Exame do Ano Pasado Solucións Exame do Ano Pasado

Unidade didáctica 6

Unidade didáctica 7

Unidade didáctica 8

Unidade didáctica 9

Unidade didáctica 10

Unidade didáctica 11

Os meus cursos

ADMINISTRACIÓN

Administración do curso

Configuración do meu perfil

[Inicio](#)[Preguntas Frecuentes](#)[Accesibilidade](#)[Aviso](#)[Privacidade](#)[Contacto](#)

