

Proyecto del Día 7

Llegó el momento de programar pensando en los principios de la programación orientada a objetos. ¿Y qué vamos a hacer hoy? Hoy te voy a pedir que crees un código que le permita a una persona realizar operaciones en su cuenta bancaria. No te asustes que la consigna va a estar bien definida para que puedas hacerlo en poco tiempo.

Primero vas a crear una clase llamada Persona, y Persona va a tener solo dos atributos: nombre y apellido. Luego, vas a crear una segunda clase llamada Cliente, y Cliente va a heredar de Persona, porque los clientes son personas, por lo que el Cliente va a heredar entonces los atributos de Persona, pero también va a tener atributos propios, como número de cuenta y balance, es decir, el saldo que tiene en su cuenta bancaria.

Pero eso no es todo: Cliente también va a tener tres métodos. El primero va a ser uno de los métodos especiales y es el que permite que podamos imprimir a nuestro cliente. Este método va a permitir que cuando el código pida imprimir Cliente, se muestren todos sus datos, incluyendo el balance de su cuenta. Luego, un método llamado Depositar, que le va a permitir decidir cuánto dinero quiere agregar a su cuenta. Y finalmente, un tercer método llamado Retirar, que le permita decidir cuánto dinero quiere sacar de su cuenta.

Una vez que hayas creado estas dos clases, tienes que crear el código para que tu programa se desarrolle, pidiéndole al usuario que elija si quiere hacer depósitos o retiros. El usuario puede hacer tantas operaciones como quiera hasta que decida salir del programa. Por lo tanto, nuestro código tiene que ir llevando la cuenta de cuánto dinero hay en el balance, y debes procurar, por supuesto, que el cliente nunca pueda retirar más dinero del que posee. Esto no está permitido.

Recuerda que ahora que sabes crear clases y objetos que son estables y que retienen información, no necesitas crear funciones que devuelvan el balance, ya que la instancia de cliente puede saber constantemente cuál es su saldo debido a que puede hacer sus operaciones llamando directamente a este atributo y no a una variable separada.

Para que tu programa funcione, puedes organizar tu código como quieras, hay muchas formas de hacerlo, pero mi recomendación es que básicamente, luego de crear las dos clases que te he mencionado, crees dos funciones una que se encarguen de crear al cliente pidiéndole al usuario toda la información necesaria y devolviendo, a través del return, un objeto cliente ya creado.

La otra función (que puede llamarse inicio, o algo por el estilo), es la función que organiza la ejecución de todo el código: primero llama a la función “crear cliente” y luego se encarga de

mantener al usuario en un loop que le pregunte todo el tiempo si quiere depositar, retirar o salir del programa y demostrarle el balance, cada vez que haga una modificación.

Para que este programa no se te haga súper largo o complejo, te propongo que esta vez no nos fijemos tanto en los controles, para ver si el usuario ha puesto opciones permitidas o no, si ha puesto números o no, si ha puesto mayúsculas o minúsculas, y creemos el código confiando en que el usuario va a ingresar siempre información apropiada. Por supuesto que si tú prefieres incluir todos esos controles, está genial.

Yo en mi solución me voy a dedicar simplemente a crear el código duro para que la explicación no se haga super larga.

¿Estás listo? Yo sí.

Ponte a programar y diviértete mucho que yo te estoy esperando en la lección siguiente para mostrarte mi solución.