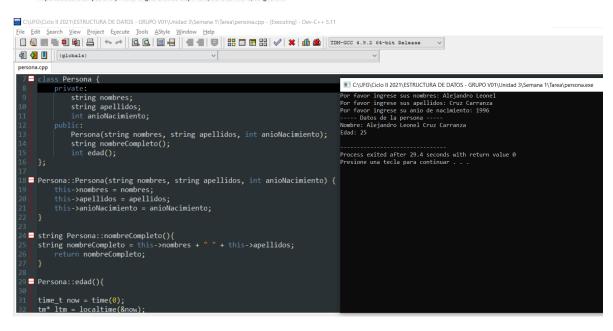
Ejercicio 1:

Cree una clase **CuentaBancaria** con tres atributos: **numeroCuenta, nombreTitular** y **saldo** (un número mayor a cero, no se permite sobregirar la cuenta). Debe asignar valores a los tres atributos en el método **constr**uctor. Además, agregue los siguientes 5 métodos:

- getSaldo: devuelve el saldo disponible en la cuenta bancaria.
- getNumeroCuenta: devuelve el numeroCuenta asociado a la cuenta bancaria
- getNombreTitular: devuelve el nombreTitular asociado a la cuenta bancaria.
- depositar: recibe el parámetro importeDeposito, que deberá sumarse al saldo de la cuenta. El
 importe deberá ser positivo; de ser negativo, deberá ignorarse.
- retirar: recibe el parámetro importeRetiro, que deberá restarse del saldo de la cuenta. El importe deberá ser positivo y menor o igual al saldo disponible; de lo contrario, se ignorará.



Ejercicio 2:

Cree una clase **Persona** con tres atributos: **nombres**, **apellidos** y **anioNacimiento**. Debe asignar valores a los tres atributos en el método **constr**uctor. Además, agregue los siguientes 2 métodos:

- nombreCompleto: debe devolver los nombres concatenados con un espacio en blanco y con los apellidos de la persona.
- **edad:** devolverá el resultado de restar el año de nacimiento del año actual (2021). Opcional: investigue cómo se puede obtener el año a partir de la fecha actual en C++.

```
(globals)
 persona.cpp cuentabancaria.cpp
                                                                                                                                                                                                                           ■ C:\UFG\Ciclo II 2021\ESTRUCTURA DE DATOS - GRUPO V01\Unidad 3\Semana 1\Tarea\cuentabancaria.exe
                                                                                                                                                                                                                           Titular: Jonathan Joesta
# Cuenta: A-480920
Saldo: $2900
       int main() {
                                                                                                                                                                                                                           Titular: Arale Guitierrez
# Cuenta: C-165426
Saldo: $ 784661
                      // Crear una cuenta bancaria
CuentaBancaria micuenta ("A-480920", "Jonathan Joestar", 2900.00);
Cout << "Iitular: " << miCuenta.getNombreTitular() << endl;
cout << "$ Cuenta: " << miCuenta.getNumeroCuenta() << endl;
cout << "Saldo: $" << miCuenta.getSaldo() << endl;
cout << "Saldo: $" << miCuenta.getSaldo() << endl;
cout << "Saldo: $" << miCuenta.getSaldo() << endl;</pre>
                                                                                                                                                                                                                           Titular: John Cena
# Cuenta: A-354698
Saldo: $0
                                                                                                                                                                                                                           Quiero retirar $84 de la cuenta A-480920
Saldo inicial: $ 2900
Saldo luego de la operacion: $2816
             // Creenado atra cuenta/
CuentaBancaria cuenta/Amigo("C-165426", "Arale Guitlernez", 784661.00);
cout << "litular: "<< cuentaAmigo.getMombrefitular() << endl;
cout << "# Guenta: "<< cuentaAmigo.getMumeroCuenta() << endl;
cout << "Saldo: $ " << cuentaAmigo.getSaldo() << endl;
cout << "cuentaAmigo.getSaldo() << endl;</pre>
                                                                                                                                                                                                                           Quiero retirar $3500 de la cuenta A-480920
Saldo inicial: $2816
! - Importe no valido
Saldo luego de la operacion: $2816
              CuentaBancaria cuentaNoValida("A-354698", "John Cena", -500.00); cout < "Titular: "<< cuentaNoValida.getNombreTitular() << endl; cout < "B Cuenta" << cuentaNoValida.getNombreTitular() << endl; cout < "B Cuenta" << cuentaNoValida.getNombreCuenta() <= cendl; cout < < "Saldo: $" << cuentaNoValida.getSaldo() << endl;
                                                                                                                                                                                                                           Quiero retirar -$725 de la cuenta A-480920
Saldo inicial: $2816
| - Importe no valido
Saldo luego de la operacion: $2816
                                                                                                                                                                                                                           Quiero depositar $125000 a la cuenta A-480920
Saldo inicial: $2816
Saldo luego de la operacion: $2816
              // Hacer un retiro de La cuenta miCuenta "<< miCuenta getHumeroCuenta() << end1;
cout << "Quiero retirar $84 de la cuenta "<< miCuenta getHumeroCuenta() << end1;</pre>
```

Ejercicio 3:

Cree una clase **Personaje** con dos atributos: **nombre** (cadena) y **puntosVida** (un número positivo). Ambos atributos deben llenarse con un constructor. Además, agregue los siguientes 5 métodos:

- recibirGolpe: recibe el parámetro puntosPerdidos, que representa la cantidad de puntos a sustraer de los puntosVida como resultado de que el personaje reciba un golpe sin defenderse.
- estaVivo: devolverá true si los puntos de vida son mayores a cero y false, en caso contrario.
- getPuntosVida: devuelve el valor del atributo privado puntosVida
- getNombre: devuelve el valor del atributo privado nombre
- comer: es lo opuesto a recibirGolpe, suma el parámetro puntosGanados a los puntos de vida del personaje, como resultado de alimentarse, si el personaje está vivo; de lo contrario, no tiene ningún efecto (no se puede "revivir" a un personaje fallecido (3)).

