

```

//Luis Gerardo Juarez Garcia - A00836928
//Amhed Cortes A00835822
#include <iostream>
using namespace std;

//Creamos las clases necesarias para el desarrollo del programa
//Asignamos los datos publicos y privados dependiendo de la clase
//Cada clase tiene una funcion especifica para hacer una evaluacion basica de una
transaccion
class tarifa{
public:
double tar;
void getTarifa(){
    Reinicio:
    try {
        cout<<"Ingrese la Tarifa: \n";
        cin>>tar;
        if (tar<=100000)
            cout<<"Tarifa ingresada: "<<tar<<"\n";
        else{
            throw (tar);
        }
    }
    catch (double tar){
        cout<<"Error \n El valor no puede ser mayor a 100000. \n";
        cout<<"El valor que ingreso fue "<<tar<<"\n";
        goto Reinicio;
    }
}
double setTarifa(){
    return tar;
}
};

```

```

//Comenzamos a heredar herencias en las clases
class TarifaDeProcesamiento: tarifa{
protected:
    double tar;
public:
    double tarpro;
    void getTarpro(double tar){
        if (tar<=33334){
            tarpro=600;
        }
        else if (tar<=666667){
            tarpro=800;
        }
    }
}

```

```

        else if (tar>=100000){
            tarpro=1000;
        }
        cout<<tarpro<<"\n";
    }
    double setTarpro(){
        return tarpro;
    }
};

class TipoDeMoneda{
public:
    double con;
    string Tipo;
    void getTipoDeMoneda(){
        Reinicio:
        cout<<"Que tipo de moneda se utilizara para la transferencia
(dolares/pesosmxn): \n";
        cin>>Tipo;
        if (Tipo=="dolares"){
            con=20;
        }
        else if (Tipo=="pesosmxn"){
            con=1;
        }
        else{
            cout<<"Lo siento, por ahora no utilizamos ese tipo de moneda. \n";
            goto Reinicio;
        }
    }
    string setTipo(){
        return Tipo;
    }
    double setCon(){
        return con;
    }
};

class Transferencia: public TipoDeMoneda, public TarifaDeProcesamiento {
private:
    double con, tar, tarpro;
public:
    double tarcalc, tart;
    void getTransferencia(double con, double tar, double tarpro){
        tarcalc=tar+tarpro;
        tart=tarcalc/con;
    }
    double setTransferencia(){
        return tart;
    }
};

```

```

    }
};
class DatosBancario{
public:
    string NDBB, DDBB, NDBD, DBB, TDCD, IBAN, SWIFT;
    void getNDBB(){
        cout<<"Ingrese el Nombre del Banco Beneficiario: \n";
        cin>>NDBB;
    }
    void getDDBB(){
        cout<<"\nIngrese la Direccion del Banco Beneficiario: \n";
        cin>>DDBB;
    }
    void getNDBD(){
        cout<<"\nIngrese el Nombre del Banco Destinatario: \n";
        cin>>NDBD;
    }
    void getDBB(){
        cout<<"\nIngrese la Direccion del Banco Destinatario: \n";
        cin>>DBB;
    }
    void getTDCD(){
        Reinicio:
        cout<<"\nIngrese el Tipo de cuenta (cheque/ahorro/empresarial): \n";
        cin>>TDCD;
        if (TDCD!="cheque" && TDCD!="ahorro" && TDCD!="empresarial"){
            cout<<"Tipo de Cuenta no valida \n";
            goto Reinicio;
        }
    }
    void getIBAN(){
        cout<<"\nIngresa el IBAN de la Cuenta Beneficiaria: \n";
        cin>>IBAN;
    }
    void getSWIFT(){
        cout<<"\nIngrese el SWIFT de la Cuenta Destinataria: \n";
        cin>>SWIFT;
    }
    string setNDBB(){
        return NDBB;
    }
    string setDDBB(){
        return DDBB;
    }
    string setNDBD(){
        return NDBD;
    }
}

```

```

string setDBB(){
    return DBB;
}
string setTDCD(){
    return TDCD;
}
string setIBAN(){
    return IBAN;
}
string setSWIFT(){
    return SWIFT;
}
};

```

//Le solicitamos al usuario informacion de la transaccion  
 //El programa hace un ticket de los datos importantes y los imprime  
 //En este caso el usuario ingresara una tarifa de 1000, y dara la moneda de dolares

//El programa devolvera una tarifa de procesamiento de 600 y un total de 80  
 int main() {

```

    int N;
    string DatosBancarios[7];
    cout<<"Bienvenido a Transferencia en linea \n";
    cout<<"Porfavor, me puede brindar los siguientes datos? \n";
    DatosBancario Datos1;
    Datos1.getNDBB();
    Datos1.getDDBB();
    Datos1.getNDBD();
    Datos1.getDBB();
    Datos1.getTDCD();
    Datos1.getIBAN();
    Datos1.getSWIFT();
    DatosBancarios[0]=Datos1.setNDBB();
    DatosBancarios[1]=Datos1.setDDBB();
    DatosBancarios[2]=Datos1.setNDBD();
    DatosBancarios[3]=Datos1.setDBB();
    DatosBancarios[4]=Datos1.setTDCD();
    DatosBancarios[5]=Datos1.setIBAN();
    DatosBancarios[6]=Datos1.setSWIFT();
    cout<<"\nDatos del Banco destinatario: \n";
    for (int i=0;i<2;i++){
        cout<<DatosBancarios[i]<<"\n";
    }
    cout<<DatosBancarios[5]<<"\n";
    cout<<"\nDatos del banco Beneficiario: \n";
    for (int i=2;i<5;i++){
        cout<<DatosBancarios[i]<<"\n";
    }
}

```

```

}
cout<<DatosBancarios[6]<<"\n";

string tarifas;
tarifa tarifa1;
tarifa1.getTarifa();
tarifa1.setTarifa();

string procesamiento;
TarifaDeProcesamiento tarifaP;
tarifaP.getTarpro(tarifa1.setTarifa());
tarifaP.setTarpro();

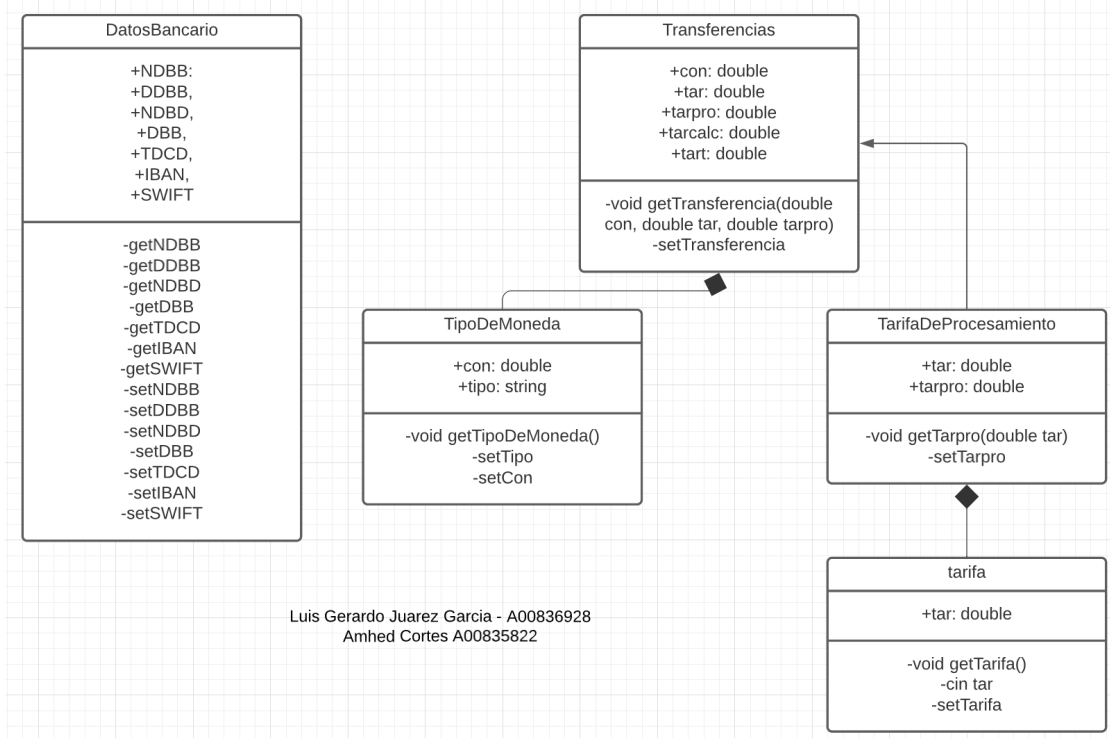
string monedas;
TipoDeMoneda moneda1;
moneda1.getTipoDeMoneda();
moneda1.setCon();
moneda1.setTipo();

string transferencias;
Transferencia trans1;
trans1.getTransferencia(moneda1.setCon(),tarifaP.setTarpro(),tarifa1.setTarifa());
trans1.setTransferencia();

cout<<"\nTransferencia en Linea \n";
cout<<"\nNombre del Banco Beneficiaria: "<<DatosBancarios[0]<<"\n";
cout<<"\nDireccion del Banco Beneficiaria: "<<DatosBancarios[1]<<"\n";
cout<<"\nNumero del IBAN de la Cuenta Beneficiaria: 
"<<DatosBancarios[5]<<"\n";
cout<<"\nNombre del Banco Destinatario: "<<DatosBancarios[2]<<"\n";
cout<<"\nDireccion del Banco Destinatario: "<<DatosBancarios[3]<<"\n";
cout<<"\nTipo de la Cuenta Destinataria: "<<DatosBancarios[4]<<"\n";
cout<<"\nNumero SWIFT de la Cuenta Destinataria: "<<DatosBancarios[6]<<"\n";
cout<<"\nTransferencia: "<<tarifa1.setTarifa()<<"\n";
cout<<"\nTarifa de procesamiento: "<<tarifaP.setTarpro()<<"\n";
cout<<"\nMoneda utilizada en la transaccion: "<<moneda1.setTipo()<<"\n";
cout<<"\nTotal: "<<trans1.setTransferencia()<<"\n";
}

```

## Diagrama:



## Casos de prueba:

Casos	Datos ingresados	Resultado esperado
1.- Limite de tarifa	Tarifa ingresada: 100001	Que el programa no le permita al usuario ingresar un valor mayor a 100000 y le obligue a ingresar una tarifa valida
2.- Moneda utilizada	euro	El programa informara al usuario que ese tipo de moneda aun no es valida.

## Referencias bibliograficas:

Hamilton, H. (2022, 5 septiembre). *Qué se necesita para una transferencia bancaria por internet*. PayRetailers. [https://payretailers.com/es/latam/lo-que-necesitas-para-hacer-una-transferencia-bancaria-por-internet\\_2/](https://payretailers.com/es/latam/lo-que-necesitas-para-hacer-una-transferencia-bancaria-por-internet_2/)

## Video demostrativo:

<https://youtu.be/ZppCMAIJR8c>