

CECS 2222-22

Actividad 1.2

12 de marzo del 2022

Código Fuente:

El código fue creado en tres archivos distintos. Un header File, uno cpp y el main. Se comenta en el código el file de referencia.

```
// Stock Class declaration header file
#ifndef STOCK_H
#define STOCK_H
#include <iostream>
using namespace std;

class Stock
{
    private:
        float shares, pps, comm;           // Shares, Price per
        Share, Commision

    public:
        Stock();                           // Default constructor
        Stock(float, float, float);        // Constructor
        ~Stock();                           // Destructor

        // This are the mutators functions
        void setShares(float);
        void setPrice(float);
        void setComm(float);
        void setAll(float, float, float);

        // This are the accesors functions
        float getShares() const;
        float getPrice() const;
        float getTruePrice() const;
        float getComm() const;
        float getCommPrice() const;
        float getTotal() const;
        void display() const;
};

#endif
```

```
// CPP file for the functions definition

#include <iostream>
#include "Stock.h"
using namespace std;

// Defining the member functions for the class Stock.
Stock::Stock()
{
    setShares(0);
    setPrice(0);
    setComm(0);
}

Stock::Stock(float aShares, float aPps, float aComm)
{
    setShares(aShares);
    setPrice(aPps);
    setComm(aComm);
}

void Stock::setAll(float aShares, float aPps, float aComm)
{
    setShares(aShares);
    setPrice(aPps);
    setComm(aComm);
}

Stock::~~Stock()
{
    cout << "\nThank you for using our services.\n";
}

// Here we define the mutators
void Stock::setShares(float aShares)
{
    shares = aShares;
}

void Stock::setPrice(float aPps)
{
    pps = aPps;
}

void Stock::setComm(float aComm)
{
    comm = aComm;
}
```

```
// Here we define the accessors
float Stock::getShares() const
{
    return shares;
}

float Stock::getPrice() const
{
    return pps;
}

float Stock::getComm() const
{
    return comm * 0.01;
}

float Stock::getTruePrice() const
{
    return getPrice() * getShares();
}

float Stock::getCommPrice() const
{
    return getTruePrice() * getComm();
}

float Stock::getTotal() const
{
    return getTruePrice() + getCommPrice();
}

void Stock::display() const
{
    cout << "\nAmount paid for the stock: $" << getTruePrice()
<< endl;
    cout << "Amount o commission: $" << getCommPrice() << endl;
    cout << "Total amount paid: $" << getTotal() << endl;
}
```

```
// Main file

#include <iostream>
#include "Stock.h"
using namespace std;

int main()
{
    Stock sts;
    float shares = 0, pricePerShare = 0, commissions = 0;

    cout << "How many shares of stock have you bought? ";
    cin >> shares;
    cout << "What was the stock's price per share? $";
    cin >> pricePerShare;
    cout << "What was the percent commission for the
transaction? ";
    cin >> commissions;
    sts.setAll(shares, pricePerShare, commissions);
    sts.display();

    return 0;
}
```

Ejecución:

```
How many shares of stock have you bought? 750  
What was the stock's price per share? $35  
What was the percent commission for the transaction? 2
```

```
Amount paid for the stock: $26250  
Amount o commission: $525  
Total amount paid: $26775
```

```
Thank you for using our services.
```

```
...Program finished with exit code 0  
Press ENTER to exit console.█
```

Tabla Descriptiva

Variable Name	Description
shares	Atributo de tipo float que almacena la cantidad de acciones compradas
pps	Atributo de tipo float que almacena el precio por acción
comm	Atributo de tipo float que almacena el porcentaje de comisión

Member Function	Description
Default Constructor	Inicializa un objeto de tipo Stock con unidades predeterminadas
Constructor con parámetros	Inicializa un objeto de tipo Stock con valores dados por el usuario
setPrice	Coloca el precio por acción dado por el usuario
setShares	Coloca la cantidad de acciones compradas dadas por el usuario
setComm	Coloca el porcentaje de comisión dado por el usuario
setAll	Inicializa un objeto de tipo Stock con valores dados por el usuario
getShares	Retorna la cantidad de acciones compradas
getPrice	Retorna el precio por cada acción
getTruePrice	Retorna el subtotal de la compra de acciones
getComm	Retorna el porcentaje de comisión
getCommPrice	Retorna el costo de la comisión
getTotal	Retorna el costo total de la compra de acciones
display	Imprime el contenido de un objeto de tipo Stock

Diagrama UML

Stock
-shares:float -pps:float -comm:float
+Stock() +Stock(aShares:float, aPps:float, aComm:float) +~Stock() +setShares(aShares:float):void +setPrice(aPps:float):void +setComm(aComm:float):void +getShares() const:float +getPrice() const:float +getTruePrice() const:float +getComm() const:float +getCommPrice() const:float +getTotal() const:float +display() const:void