Практична робота.

Тема: Об’єкти і класи. Інкапсуляція (приховування даних). Абстрагування. Конструктор. Деструктор. Оператор визначення діапазону доступу. Оператор належності.

Мета: Отримати практичні навички при роботі з класами та об’єктами.

*Задача.*

*Постановка задачі.*

Потрібно створити клас, який представляє пакет акцій. З цим пакетом можна виконувати такі операції:

* отримати пакет у компанії;
* продати пакет;
* отримати додаткові акції того ж пакету;
* коректувати середню вартість однієї акції пакету;
* відображати дані про пакет акцій.

Крім того нам потрібна така інформація:

* + назва компанії;
	+ число акцій у пакеті;
	+ ціна кожної акції;
	+ загальна вартість всіх акцій пакету.

Ключові слова та терміни:

* + 1. Клас – тип визначений користувачем.
		2. Інкапсуляція – приховування даних. Закритий розділ (private) ідентифікує члени класу, до яких можна доступитись через загальнодоступні функції-елементи. Така ізоляція даних від безпосереднього доступу називається приховуванням даних.
		3. Загальнодоступний або відкритий розділ – public.
		4. Абстрагування – це ідентифікація членів класу відкритого розділу.
		5. Конструктор будує об’єкт.
		6. Деструктор його руйнує.
		7. Метод ( функція-елемент класу )може бути викликаний об’єктом класу через оператор належності (.)
		8. При визначенні функції-елементу для ідентифікації класу, якому ця функція належить, використовується оператор визначення діапазону доступу (::).

Програма stock.cpp

#include <iostream>

#include <cstring> // щоб використати функцію strncpy()

using namespace std;

class Stock // Stock – ім’я класу. Stock – акція.

{

public: // відкритий розділ

 Stock(); // конструктор

 void asquire(const char \* co, int n, double pr); // отримати пакет у компанії

 void buy(int num, double price); //купити

 void sell(int num,double price); //продати

 void update(double price); //корекція вартості однієї акції

 void show(); //відобразити доні про пакет акцій

 ~Stock();// деструктор

private:

 char company[30];// назва компанії

 int shares; // кількість акцій

 double share\_val;// вартість однієї

 double total\_val;// загальна вартість

 void set\_tot() // функція для визначення загальної вартості

 {

 total\_val=shares\*share\_val;

 }

};

Stock::Stock()

{

}

Stock::~Stock()

{

}

void Stock::asquire(const char \* co, int n, double pr)

{

 strncpy(company,co,29);

 company[29]='\0';

 if (n<0)

 {

// Кількість акцій не може бути від’ємною. Встановити кількість акцій на нуль.

 cout<<"Number of shares can't be negative;"

 <<"shares set to 0\n.";

 shares=0;

 }

 else

 {

 shares=n;

 }

 share\_val=pr;

 set\_tot();

}

void Stock::buy(int num, double price)

{

 if (num<0)

 { // Число куплених акцій не може бути від’ємним. Угоду перервано.

 cout<<"Number of shares purchased can't be negative."

 <<"Transaction is aborted.\n";

 }

 else

 {

 shares+=num;

 share\_val=price;

 set\_tot();

 }

}

void Stock::sell(int num, double price)

{

 if (num<0)

 {//

 cout<<"Number of shares sold can't be negative."

 <<"Transaction is aborted.\n";

 }

 else

 {

 if(num>shares)

 {

 cout<<"You can't sell more than you have!\n;"

 <<"Transaction is aborted,\n";

 }

 else

 {

 shares-=num;

 share\_val=price;

 set\_tot();

 }

 }

}

 void Stock::show()

 {

 cout<<"Company: "<<company

 <<" Shares: "<<shares<<"\n"

 <<" Shares Price: $"<<share\_val

 <<" Total worth: $"<<total\_val

 <<"\n";

 }

 int main()

 {

 Stock stock1;

 stock1.asquire("NanoSmart",20,12.50);

//Команди форматування:

 cout.setf(ios\_base::fixed); // використовується десятковий формат //фіксованої довжини

 cout.precision(2);//в десятковому числі справа від точки може //знаходитись тільки два символи

 cout.setf(ios\_base::showpoint); // показувати заключні нулі

 stock1.show(); // відобразити дані про пакет акцій

 stock1.buy(15,18.29); // купляємо

 stock1.show();// показати дані

 stock1.sell(400,20.00); // продаємо

 stock1.show(); // показати змінені дані

 return 0;

 }

*Завдання: створіть окремо заголовний файл та файл реалізації .cpp.*