

Κώδικας Κλάσης *Products.hpp*

```
//Class Products.hpp

#include<iostream>
#include<string.h>
using namespace std;

class Products{ //Ορισμός κλάσης
private:
    int code;
    string name;
    float purchase_price, final_price, rate;

public:
    //constructor χωρίς όρισμα
    Products(){
        purchase_price = 0;
    }

    //constructor με όρισμα
    Products(float pp){
        purchase_price = pp;
    }

    //Destructor
    ~Products(){
    }

    /* Αλληλεπιδραστική μέθοδος για την εισαγωγή
    των δεδομένων του καθε είδους */
    void readProductData(){
        cout << "==== Entering Values =====" << endl;
        cout << "Give Product code:"; cin >> code;
        cout << "Give Product name:"; cin >> name;
        cout << "Give Purchase Price:"; cin >> purchase_price;
        cout << "Give Profit Percentage (%):"; cin >> rate;
    }

    /* Μέθοδος για την εμφάνιση των δεδομένων των ειδών
    (με κλήση των μεθόδων υπολογισμού της τιμή πώλησης με το ποσοστό κέρδους
    και της μεθόδου του υπολογισμού της τελικής τιμής με ΦΠΑ 24%). */
    void printProductData(){
        float salesPrice;
        cout << "PRODUCT CODE: " << code << endl;
        cout << "PRODUCT NAME: " << name << endl;
        cout << "PURCHASE PRICE: " << purchase_price << " EURO" << endl;

        // Υπολογισμός τιμής με ποσοστό κέρδους προ ΦΠΑ 24%
        salesPrice = calculateProfitPrice(purchase_price, rate);

        // Υπολογισμός τελικής τιμής με ΦΠΑ 24%
        final_price = calculateFinalPrice(salesPrice);

        cout << "SALE PRICE WITHOUT VAT (24%): " << salesPrice << " EURO" << endl;
        cout << "FINAL PRICE WITH VAT (24%): " << final_price << " EURO" << endl;
        cout << "-----" << endl;
    }

    /* Μέθοδος υπολογισμού της τιμής πώλησης
    βάσει του ποσοστού κέρδους */
    float calculateProfitPrice(float _price, float _rate){
        return _price += _price * (_rate / 100);
    }

    // Μέθοδος υπολογισμού του ΦΠΑ
```

```

float calculateFinalPrice(float aPrice){
    return aPrice += (aPrice * 24) / 100;
}

// Μέθοδος επιστροφής της τιμής αγοράς
float getPurchasePrice(){
    return purchase_price;
}

// Μέθοδος επιστροφής της τελικής τιμής
float getFinalPrice(){
    return final_price;
}

/* Μέθοδος σύγκρισης τιμών προϊόντων
με αντικείμενο ως όρισμα */
bool isMoreExpensiveThan(Products aProduct){
    /* Ελέγχεται αν η τελική τιμή του αντικειμένου (προϊόντος) που καλεί τη μέθοδο
είναι μεγαλύτερη από την τελική τιμή του αντικειμένου (προϊόντος)
που βρίσκεται ως όρισμα της μεθόδου */
    if(final_price > aProduct.final_price)
        return true;
    else
        return false;
}

/* Μέθοδος σύγκρισης τιμών προϊόντων με αντικείμενα ως ορίσματα
και επιστροφή αντικειμένου. Η μέθοδος είναι τύπου Products */
Products findExpensive(Products p1,Products p2,Products p3){
    if((p1.final_price > p2.final_price) && (p1.final_price > p3.final_price))
        return p1;
    else if ((p2.final_price > p1.final_price) && (p2.final_price > p3.final_price))
        return p2;
    else
        return p3;
}
} }; //Τέλος κλάσης

```

Κυρίως πρόγραμμα *Askhsh5.cpp*

Χρήση μεθόδου αποστολής αντικειμένου ως όρισμα.

```

//Askhsh 5
//Main Programm

#include<iostream>
#include "Products.hpp" //Συμπεριλαμβάνεται η κλάση Products.hpp
using namespace std;

int main(){
    Products Prod1,Prod2,Prod3; //Τα ζητούμενα προϊόντα
    Products mostExpensive; //Το ακριβότερο προϊόν

    // Συλλογή τιμών για τα αντικείμενα
    cout << "\n+++++ PRODUCT 1 DATA INSERTION +++++" << endl;
    Prod1.readProductData();
    cout << "\n+++++ PRODUCT 2 DATA INSERTION +++++" << endl;
    Prod2.readProductData();
    cout << "\n+++++ PRODUCT 3 DATA INSERTION +++++" << endl;
    Prod3.readProductData();

    // Εκτύπωση των πληροφοριών των αντικειμένων
    cout << "\n+++++ PRINT PRODUCTS DATA +++++" << endl;
    Prod1.printProductData();
    Prod2.printProductData();
    Prod3.printProductData();
}

```

```

/* Αρχίζει η σύγκριση των προϊόντων
για να βρεθεί το ακριβότερο προϊόν */
cout << "\nThe Product Comparison to find the most expensive product begins!!!" << endl;

//Έλεγχος τελικής τιμής των Prod1 και Prod2
if(Prod1.isMoreExpensiveThan(Prod2)){
    mostExpensive = Prod1; // Το Prod1 είναι το ακριβότερο προϊόν
}
else{
    mostExpensive = Prod2; // Το Prod2 είναι το ακριβότερο προϊόν
}

/* Έλεγχος τελικής τιμής του πιο ακριβού από τον
προηγούμενο έλεγχο (mostExpensive) και του Prod3 */
if(Prod3.isMoreExpensiveThan(mostExpensive)){
    mostExpensive = Prod3; // Το Prod3 είναι το ακριβότερο προϊόν
}

/* Εμφάνιση των στοιχείων του
ακριβότερου από τα τρία προϊόντα */
cout << "\nThe most expensive product is" << endl;
mostExpensive.printProductData();

return 0;
}

```

Κυρίως πρόγραμμα *Askhsh5_2.cpp*

Χρήση μεθόδου αποστολής αντικειμένου ως όρισμα και επιστροφής αντικειμένου από την μέθοδο.

```

//Askhsh 5_2
//Main Programm

#include<iostream>
#include "Products.hpp" //Συμπεριλαμβάνεται η κλάση Products.hpp
using namespace std;

int main(){
    Products Prod1,Prod2,Prod3; //Τα ζητούμενα προϊόντα
    Products agent, mostExpensive; //Χρήση βοηθού και ακριβότερο προϊόν

    // Συλλογή τιμών για τα αντικείμενα
    cout << "\n+++++ PRODUCT 1 DATA INSERTION +++++" << endl;
    Prod1.readProductData();
    cout << "\n+++++ PRODUCT 2 DATA INSERTION +++++" << endl;
    Prod2.readProductData();
    cout << "\n+++++ PRODUCT 3 DATA INSERTION +++++" << endl;
    Prod3.readProductData();

    // Εκτύπωση των πληροφοριών των αντικειμένων
    cout << "\n+++++ PRINT PRODUCTS DATA +++++" << endl;
    Prod1.printProductData();
    Prod2.printProductData();
    Prod3.printProductData();

    /* Αρχίζει η σύγκριση των προϊόντων
για να βρεθεί το ακριβότερο προϊόν */
    cout << "\nThe Product Comparison to find the most expensive product begins!!!" << endl;

    mostExpensive = agent.findExpensive(Prod1,Prod2,Prod3);

    /* Εμφάνιση των στοιχείων του
ακριβότερου από τα τρία προϊόντα */
    cout << "\nThe most expensive product is" << endl;
}

```

```
mostExpensive.printProductData();  
return 0;  
}
```