**Ανοικτά Ακαδημαϊκά Μαθήματα**

**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας**

**Αντικειμενοστρεφής Προγραμματισμός (Ε)**

**Ενότητα 3:** Κληρονομικότητα - Λύσεις

Κλειώ Σγουροπούλου

Τμήμα Μηχανικών Πληροφορικής ΤΕ

|  |  |
| --- | --- |
| Το περιεχόμενο του μαθήματος διατίθεται με άδεια Creative Commons εκτός και αν αναφέρεται διαφορετικά | Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους. |

# Αντικειμενοστρεφής Προγραμματισμός- Εργαστήριο 3- Λύσεις

## **Συνολική λύση άσκησης 1 και 3**

#include <iostream>

#include <string>

using namespace std;

class product

{

protected:

string name;

string producer;

float price;

public:

product(string \_name, string \_producer, float \_price)

{

name = \_name; producer = \_producer; price = \_price;

}

product()

{

name = "N/A";

producer = "N/A";

price = 0;

}

void display()

{

cout << "Product name: " << name << endl;

cout << " Producer: " << producer << endl; cout << " Price: " << price << endl;

}

};

class wine: public product

{

protected:

string variety;

public:

//είναι καλύτερο να παραπέμπουμε την αρχικοποίηση των 3 κοινών παραμέτρων

(name, producer, price)

//από τον κατασκευαστή της μητρικής κλάσης

wine(string \_name, string \_producer, float \_price, string

\_variety):product(\_name, \_producer, \_price)

{

variety = \_variety;

}

wine()

{

variety = "N/A";

}

void display()

{

cout << "Wine name: " << name << endl;

cout << " Producer: " << producer << endl; cout << " Price: " << price << endl;

cout << " Variety: " << variety << endl;

}

};

class spirit: public product

{

protected:

string type;

public:

spirit(string \_name, string \_producer, float \_price, string

\_type):product(\_name, \_producer, \_price)

{

type = \_type;

}

spirit()

{

type = "N/A";

}

void display()

{

cout << "Spirit name: " << name << endl;

cout << " Producer: " << producer << endl;

cout << " Price: " << price << endl;

cout << " Type: " << type << endl;

}

};

int main(int argc, char \* argv[])

{

spirit s("Absolut", "Absolute Vodkas SA", 23.5, "Vodka");

s.display();

wine w; w.display();

system("pause");

return 0;

}

## **Λύση άσκησης 2**

Αρχικός κώδικας:

#include <iostream>

using namespace std;

class A

{

private:

int var\_a;

void show();

public:

void check\_B();

};

class B : public A

{

public:

int var\_b;

void display();

};

void A::show()

{

cout << var\_a << endl;

}

void A::check\_B()

{

cout << var\_b << endl;

}

void B::display()

{

cout << var\_a << ' ' << var\_b << ' ' << endl;

}

int main()

{

B b;

cin >> b.var\_b; cin >> b.var\_a;

b.show();

b.display();

return 0;

}

Τα λάθη στον κώδικα είναι:

Α) Η μεταβλητή var\_b είαι ορισμένη στην παραγόμενη κλάση Β, συνεπώς η κλάση Α (ως μητρική) δεν γνωρίζει τίποτα για αυτήν. Μια λύση θα ήταν να ανεβάσουμε την μεταβλητή στην κλάση Α (ως protected).

Β)Η μεταβλητή var\_a δεν είναι ορατή από την κλάση Β γιατί είναι private στην κλάση Α. Λύση: την μετατρέπουμε σε protected.

Γ)Στην main, οι μεταβλητές var\_a, var\_b και η συνάρτηση show δεν είναι ορατές. Λύση: δημιουργία συναρτήσεων setter (setA, setB) και μετατροπή της show σε public.

Τελικός κώδικας

#include <iostream>

using namespace std;

class A

{

private:

protected:

int var\_a;

int var\_b;

public:

void show();

void check\_B(); void setA(int a); void setB(int b);

};

class B : public A

{

public:

void display();

};

void A::show()

{

cout << var\_a << endl;

}

void A::check\_B()

{

cout << var\_b << endl;

}

void A::setA(int a)

{

var\_a = a;

}

void A::setB(int b)

{

var\_b = b;

}

void B::display()

{

cout << var\_a << ' ' << var\_b << ' ' << endl;

}

int main()

{

B b;

int temp\_b, temp\_a; cin >> temp\_b;

cin >> temp\_a; b.setB(temp\_b); b.setA(temp\_a);

b.show();

b.display();

return 0;

}

|  |
| --- |
| **Ανοικτά Ακαδημαϊκά Μαθήματα****Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας** |
| **Τέλος Ενότητας** |
| **Χρηματοδότηση*** Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
* Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Αθήνας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
* Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

 |

**Σημειώματα**

**Σημείωμα Αναφοράς**

Copyright ΤΕΙ Αθήνας, Κλειώ Σγουροπούλου, 2014. Κλειώ Σγουροπούλου. «Αντικειμενοστρεφής Προγραμματισμός». Ενότητα 3: «Κληρονομικότητα –Λύσεις ασκήσεων». Έκδοση: 1.0. Αθήνα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: [ocp.teiath.gr](https://ocp.teiath.gr/).

**Σημείωμα Αδειοδότησης**

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] http://creativecommons.org/licenses/by-nc-sa/4.0/

Ως **Μη Εμπορική** ορίζεται η χρήση:

* που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
* που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
* που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

**Διατήρηση Σημειωμάτων**

* Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
* Το Σημείωμα Αναφοράς
* Το Σημείωμα Αδειοδότησης
* Τη δήλωση Διατήρησης Σημειωμάτων
* Το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει) μαζί με τους συνοδευόμενους υπερσυνδέσμους.