



Ελληνική Δημοκρατία
Τεχνολογικό Εκπαιδευτικό
Ίδρυμα Ηπείρου

Αντικειμενοστραφής Προγραμματισμός

Ενότητα 12 : Επαναληπτικές Ασκήσεις (3/3)

Ιωάννης Τσούλος



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Τμήμα Μηχανικών Πληροφορικής Τ.Ε

Αντικειμενοστραφής Προγραμματισμός

Ενότητα 11 : Επαναληπτικές Ασκήσεις (3/3)

Ιωάννης Τσούλος

Επίκουρος Καθηγητής

Άρτα, 2015





Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.





Χρηματοδότηση

- 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.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Άσκηση (Κατηγορία χρονικών στιγμών)

Να γραφεί κατηγορία για την αναπαράστασή χρονικών στιγμών (δευτερόλεπτο, λεπτό, ώρα). Να έχει και τελεστές μοναδιαίας αύξησής δευτερολέπτων και μείωσής καθώς και έναν φιλικό τελεστή $>$ για την σύγκρισή δύο χρονικών στιγμών.



Λύση

```
1 # include <iostream>
2 using namespace std;
3
4 class Time
5 {
6     private:
7         int seconds , minutes , hours ;
8         void nextSecond ();
9         void previousSecond ();
10    public:
11        Time(int h,int m,int s);
12        void details ();
13        Time& operator++();
14        Time& operator--();
15        Time& operator++(int);
16        Time& operator--(int);
17        friend bool operator>(Time& a,Time& b);
18 };
19
20 Time :: Time (int h , int m, int s )
21 {
22     hours=h;
23     minutes=m;
24     seconds=s;
25 }
26
27 void    Time :: details ()
28 {
29     cout<<hours<<" : "<<minutes<<" : "<<seconds<<endl ;
```



Λύση

```
30 }
31
32 void Time::nextSecond()
33 {
34     ++seconds;
35     if (seconds==60)
36     {
37         seconds=0;
38         minutes++;
39         if (minutes==60)
40         {
41             minutes=0;
42             hours++;
43             if (hours==24)
44             {
45                 hours=0;
46             }
47         }
48     }
49 }
50
51 void Time::previousSecond()
52 {
53     --seconds;
54     if (seconds < 0)
55     {
56         seconds = 59;
57         --minutes;
58         if (minutes < 0)
59         {
60             minutes= 59;
61             hours --;
62             if (hours < 0) hours = 23;
63         }
64     }
65 }
66
67 Time& Time::operator++()
68 {
69     nextSecond();
70     return *this;
71 }
72
73 Time& Time::operator--()
74 {
75     previousSecond();
```



Λύση

```
76         return *this;
77     }
78
79     Time&   Time::operator++(int)
80     {
81         nextSecond();
82         return *this;
83     }
84
85     Time&   Time::operator--(int)
86     {
87         previousSecond();
88         return *this;
89     }
90
91     bool operator>(Time& a, Time& b)
92     {
93         if(a.hours>b.hours) return true;
94         else
95             if(b.hours>a.hours) return false;
96         else
97             {
98                 if(a.minutes>b.minutes) return true;
99                 else
100                     if(b.minutes>a.minutes) return false;
101                 else
102                     {
103                         if(a.seconds>=b.seconds) return true;
104                         else
105                             return false;
106                     }
107             }
108     }
109
110     int main()
111     {
112         Time t1(23,59,59);
113         t1.details();
114         t1++;
115         t1.details();
116         t1--;
117         t1.details();
118         Time t2(23,59,58);
119         t2.details();
120         ++t2;
121         t2.details();
```




Λύση

```
122     —t2;  
123     t2.details();  
124     cout <<(t1>t2)<<endl;  
125     return 0;  
126 }
```



Άσκηση (Μάντρα οχημάτων)

Δημιουργήστε μια κατηγορία για την περιγραφή οχημάτων με ιδιωτικά πεδία μάρκα και τιμή.

Να προστεθεί μια υπερβατική μέθοδος για την περιγραφή του οχήματος, ένας φιλικός τελεστής σύγκρισής $>$ ανάμεσα σε δύο οχήματα που επιστρέφει αληθές αν το πρώτο όχημα είναι πιο ακριβό από το δεύτερο και ένα στατικό πεδίο για την καταμέτρησή των οχημάτων.



Άσκηση (Μάντρα οχημάτων)

Στην συνέχεια δημιουργήστε την κατηγορία για την περιγραφή ΙΧ με πρόσθετο ιδιωτικό πεδίο το όνομα του μοντέλου καθώς και μια κατηγορία για την περιγραφή φορτηγών με πρόσθετο ιδιωτικό πεδίο το φορτίο.

Στη *main()* φτιάξτε έναν πίνακα οχημάτων και διαβάστε από το πληκτρολόγιο το είδος του οχήματος, τιμή μάρκα και αν απαιτείται μοντέλο ή φορτίο.

Τα οχήματα να μπουν σε έναν δυναμικό πίνακα *vector*.



Άσκηση (Μάντρα οχημάτων)

Το πρόγραμμα να διαγράψει από τον πίνακα όσα αυτοκίνητα έχουν τιμή κάτω από 1000 ευρώ (λανθασμένη είσοδος) και να εμφανίζεται στο τέλος το πλήθος των συνολικών οχημάτων.

```
1 # include <iostream>
2 # include <string>
3 # include <vector>
4 using namespace std;
5
6 class Vehicle
7 {
8     protected:
9         string brand;
10        double price;
11    public:
12        static int counter;
13
14        Vehicle(string b,double p);
15
16        virtual void details()=0;
17        double getPrice ()
18        {
19            return price;
20        }
21        friend bool operator>(Vehicle &a,Vehicle &b);
22        ~Vehicle ();
23 };
24
```



Λύση

```
25 int Vehicle::counter=0;
26
27 Vehicle::Vehicle(string b,double p)
28 {
29     brand=b;
30     price=p;
31     ++counter;
32 }
33
34 bool operator>(Vehicle &a,Vehicle &b)
35 {
36     if(a.price>b.price) return true;
37     else
38     return false;
39 }
40
41 Vehicle::~~Vehicle()
42 {
43     --counter;
44 }
45
46 class Auto :public Vehicle
47 {
48     private:
49         string model;
50     public:
51         Auto(string m,string b,int p);
52         virtual void details();
53 };
54
55 Auto::Auto(string m, string b,int p)
56     : Vehicle(b,p)
57 {
58     model=m;
59 }
60
61 void Auto::details()
62 {
63     cout<<"Auto_details\n";
64     cout<<"Model:_"<<model<<endl;
65     cout<<"Brand:_"<<brand<<endl;
66     cout<<"Price:_"<<price<<endl;
67 }
68
69 class Truck :public Vehicle
70 {
```



Λύση

```

71         private:
72             double cargo;
73         public:
74             Truck(double c,string b,int p);
75             virtual void details();
76     };
77
78
79     Truck::Truck(double c,string b,int p)
80         : Vehicle(b,p)
81     {
82         cargo=c;
83     }
84
85     void    Truck::details()
86     {
87         cout<<"Truck_details\n";
88         cout<<"Cargo:_"<<cargo<<endl;
89         cout<<"Brand:_"<<brand<<endl;
90         cout<<"Price:_"<<price<<endl;
91     }
92
93     int main()
94     {
95         vector<Vehicle* > cars;
96         int totalcars=5;
97         int i;
98         for(i=0;i<totalcars;i++)
99             {
100                 Vehicle *v;
101                 int type;
102                 string brand;
103                 double price;
104                 cout<<"Enter_car_type_(1-_auto ,_2-_truck_)_";
105                 cin>>type;
106                 if(type==1)
107                     {
108                         string model;
109                         cout<<"Enter_model\n";
110                         cin>>model;
111                         cout<<"Enter_brand_\n";
112                         cin>>brand;
113                         cout<<"Enter_price\n";
114                         cin>>price;
115                         v=new Auto(model,brand,price);
116                     }

```



Λύση

```
117         else
118         {
119             double cargo;
120             cout<<"Enter_cargo_\n";
121             cin>>cargo;
122             cout<<"Enter_brand_\n";
123             cin>>brand;
124             cout<<"Enter_price_\n";
125             cin>>price;
126             v=new Truck(cargo,brand,price);
127         }
128         cars.push_back(v);
129         v->details();
130     }
131     for(i=0;i<totalcars;i++)
132     {
133         if(cars[i]->getPrice()<1000)
134         {
135             delete cars[i];
136             cars[i]=NULL;
137         }
138     }
139     cout<<"Total_Cars_are_"<<Vehicle::counter<<endl;
140     return 0;
141 }
```



Άσκηση (Αρχείο δύο στηλών)

Στο αρχείο *in.txt* υπάρχουν δύο στήλες αριθμών. Στην πρώτη στήλη είναι ο μισθός ενός υπαλλήλου και στην δεύτερη το *bonus* που θα πάρει για την νέα χρονιά. Στην τελευταία γραμμή υπάρχει η καταχώρηση 0 0. Οι δύο αριθμοί χωρίζονται μεταξύ τους με κενό.

Να γραφεί πρόγραμμα το οποίο θα διαβάζει αυτά τα αρχεία, θα αποθηκεύει αυτές τις καταχωρήσεις σε ένα *vector* από αντικείμενα της κατηγορίας *Employee*, θα ταξινομεί αυτόν τον πίνακα με βάση τις συνολικές αποδοχές των υπαλλήλων και θα εμφανίζει στο τέλος τους δύο υπαλλήλους με τις υψηλότερες αποδοχές.



Λύση

```
1 # include <iostream>
2 # include <fstream>
3 # include <string>
4 # include <vector>
5 using namespace std;
6
7 class Employee
8 {
9     private:
10         double salary;
11         double bonus;
12     public:
13         Employee();
14         void setSalary(double s);
15         void setBonus(double b);
16         void details();
17         friend bool operator>(Employee &a,Employee &b);
18 };
19
20 Employee::Employee()
21 {
22
23     salary=0.0;
24     bonus=0.0;
25 }
26
27
28 void Employee::setSalary(double s)
29 {
30     salary=s;
31 }
32
33 void Employee::setBonus(double b)
34 {
35     bonus=b;
36 }
37
38 void Employee::details()
39 {
40     cout<<salary<<" "<<bonus<<endl;
41 }
42
43 bool operator>(Employee &a,Employee &b)
44 {
45     double total1=a.salary+a.bonus;
46     double total2=b.salary+b.bonus;
47     if(total1>total2) return true;
48     else return false;
49 }
50
51 int main()
52 {
53     int i,j;
54     vector<Employee> p;
```



Λύση

```
55     ifstream infile("in.txt");
56     while(true)
57     {
58         double s,b;
59         infile >>s>>b;
60         if(s==0.0 && b==0.0) break;
61         Employee n;
62         n.setSalary(s);
63         n.setBonus(b);
64         p.push_back(n);
65     }
66
67     infile.close();
68
69     for(i=0;i<p.size();i++)
70     {
71         for(j=0;j<p.size()-1;j++)
72         {
73             if(p[j+1]>p[j])
74             {
75                 Employee temp=p[j];
76                 p[j]=p[j+1];
77                 p[j+1]=temp;
78             }
79         }
80     }
81     p[0].details();
82     p[1].details();
83     return 0;
84 }
```



Βιβλιογραφία

1. Εγχειρίδιο της C++, 2η Ελληνική έκδοση, Jesse Liberty, Γκιούρδας.
2. Μάθετε τη C++, 2η Ελληνική έκδοση, Jesse Liberty , Γκιούρδας.
3. Προγραμματισμός με τη γλώσσα C++ Μέρος Α, Αλεβίζος Θ., Έκδοση ΤΕΙ Καβάλας
4. C++ Αντικειμενοστραφής Προγραμματισμός Υπολογιστών Τομαράς Α., , Εκδόσεις Νέων Τεχνολογιών.
5. Ανακαλύψτε τη γλώσσα C, J. Purdum, Εκδόσεις Δίαυλος.
6. Εισαγωγή στο Συστηματικό Προγραμματισμό και στη γλώσσα C++, Σ. Μπαλτζής, εκδόσεις πανεπιστημίου Ιωαννίνων.
7. C++ From the beginning, Jan Skansholm, Addison Wesley.
8. The design and analysis of computer algorithms, A.V. AHO, J.E. HOPCROFT, J.D. ULLMANN, Addison Wesley 1974.
9. Structure and Interpretation of Computer Programs, H. ABELSON, G.J. SUSSMAN, J. SUSSMAN, MIT Press, Mc Graw Hill Book Company, 1985
10. The art of computer programming, D.E. KNUTH, Addison-Wesley.



Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Ιωάννης Τσούλος.

Αντικειμενοστραφής Προγραμματισμός.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή
διεύθυνση:

<http://eclass.teiep.gr/courses/COMP113/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. 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».



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

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης
Άρτα, 2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Τέλος Ενότητας

Επαναληπτικές Ασκήσεις (3/3)



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

