



Ελληνική Δημοκρατία
Τεχνολογικό Εκπαιδευτικό
Ίδρυμα Ηπείρου

Αντικειμενοστραφής Προγραμματισμός

Ενότητα 10 : Επαναληπτικές Ασκήσεις (1/3)

Ιωάννης Τσούλος



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Τμήμα Μηχανικών Πληροφορικής Τ.Ε

Αντικειμενοστραφής Προγραμματισμός

Ενότητα 10 : Επαναληπτικές Ασκήσεις (1/3)

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 1

Να δημιουργηθεί μια κατηγορία με το όνομα **Author** για συγγραφείς. Στα ιδιωτικά της πεδία να περιλαμβάνονται τα ακόλουθα:

1. Επίθετο συγγραφέα
2. Έτος γέννησής
3. Έτος θανάτου (αν υπάρχει, αλλιώς 0)

Στα δημόσιά πεδία θα περιλαμβάνονται

1. Μια μέθοδος δημιουργίας
2. Μέθοδοι *set*, *get*
3. Μια μέθοδος με το όνομα *details()* που θα εμφανίζει τα στοιχεία του συγγραφέα.

(ΣΥΝΕΧΕΙΑ →)



Άσκηση 1

Να δημιουργηθεί μια κατηγορία για την περιγραφή βιβλίων με το όνομα **Book**. Η μέθοδος να περιλαμβάνει τα ακόλουθα ιδιωτικά πεδία:

1. Όνομα βιβλίου
2. Συγγραφέας
3. Σελίδες

Στα δημόσιά πεδία να υπάρχουν:

1. Δύο μέθοδοι δημιουργίας της επιλογής σας
2. Μέθοδοι *set()* και *get()*
3. Μια μέθοδος με το όνομα *details()* που θα εμφανίζει πληροφορίες για το βιβλίο

(ΣΥΝΕΧΕΙΑ →)



Άσκηση 1

Στην συνέχεια να γραφεί κατηγορία με το όνομα **Library**. Η κατηγορία αυτή θα έχει στα ιδιωτικά της πεδία τα ακόλουθα

1. Έναν πίνακα από 100 δείκτες σε *Book*
2. Έναν μετρητή που θα καταγράφει πόσα βιβλία υπάρχουν (το πολύ 100)

Στα δημόσια πεδία θα υπάρχουν τα ακόλουθα

1. Μια μέθοδος δημιουργίας της επιλογής σας
2. Μια μέθοδος με το όνομα *int Register(har *title, Author *a, int npages)*, η οποία θα εισάγει ένα νέο βιβλίο στην βιβλιοθήκη. Αν μπορεί να γίνει εισαγωγή η μέθοδος θα επιστρέφει 1, αλλιώς 0. (ΣΥΝΕΧΕΙΑ →)



Άσκηση 1

3. Μια μέθοδος με το όνομα *int nbooks(har *a)*, η οποία θα εμφανίζει πόσα βιβλία έχουν γραφεί από τον συγκεκριμένο συγγραφέα.
4. Μια μέθοδος με το όνομα *int lend(har *title)*, η οποία θα δανείζει το βιβλίο. Αν το βιβλίο δεν μπορεί να δανειστεί, τότε η μέθοδος θα επιστρέφει 0. Για να επιστρέψει πάλι το βιβλίο πίσω θα πρέπει να χρησιμοποιηθεί η μέθοδος *register*.
5. Μια μέθοδος με το όνομα *details()*, η οποία θα εμφανίζει όλα τα βιβλία της βιβλιοθήκης.



Λύση

```
1 # include <string.h>
2 # include <iostream>
3 using namespace std;
4
5 class Author
6 {
7     private:
8         char *name,*lastname;
9         int birthyear;
10        int deathyear;
11    public:
12        Author();
13        Author(char *n,char *l,int b,int d);
14        const char *getname();
15        const char *getlastname();
16        int getBirthYear();
17        int getDeathYear();
18        void details();
19        ~Author();
20 };
21
22 Author::Author()
23 {
24     name=new char[2];
25     lastname=new char[2];
```



Λύση

```
26     strcpy (name, "");
27     strcpy (lastname, "");
28     birthyear=0;
29     deathyear=0;
30 }
31
32 Author::Author(char *n, char *l, int b, int d)
33 {
34     name=new char[strlen(n)+1];
35     strcpy (name,n);
36     lastname=new char[strlen(l)+1];
37     strcpy (lastname,l);
38     birthyear=b;
39     deathyear=d;
40 }
41
42 const char *Author::getname ()
43 {
44     return name;
45 }
46
47 const char *Author::getlastname ()
48 {
49     return lastname;
50 }
51
52 int Author::getBirthYear ()
53 {
54     return birthyear;
55 }
56
57 int Author::getDeathYear ()
58 {
59     return deathyear;
60 }
61
62 void Author::details ()
63 {
64     cout<<" Author_details====\n";
65     cout<<"Name: _"<<name<<endl;
66     cout<<"Lastname: _"<<lastname<<endl;
67     cout<<birthyear<<"-";
68     if (deathyear!=0) cout<<deathyear;
69     cout<<endl;
70 }
71
```



Λύση

```
72 Author::~Author()
73 {
74     delete[] name;
75     delete[] lastname;
76 }
77
78 class Book
79 {
80     private:
81         Author *writer;
82         char *title;
83         int npages;
84     public:
85         Book(Author *w,char *t,int n);
86         const char *getTitle();
87         const Author *getAuthor();
88         const int getPages();
89         void details();
90         ~Book();
91 };
92
93 Book::Book(Author *w,char *t,int n)
94 {
95     writer = w;
96     title=new char[strlen(t)+1];
97     strcpy(title,t);
98     npages = n;
99 }
100
101 const char *Book::getTitle()
102 {
103     return title;
104 }
105
106 const Author *Book::getAuthor()
107 {
108     return writer;
109 }
110
111 const int Book::getPages()
112 {
113     return npages;
114 }
115
116 void Book::details()
117 {
```



Λύση

```
118         cout<<"Book_details====\n";
119         cout<<"Book_title_:_"<<title<<endl;
120         cout<<"Book_pages_:_"<<npages<<endl;
121         writer->details();
122     }
123
124     Book::~Book()
125     {
126         delete[] title;
127     }
128
129     class Library
130     {
131     private:
132         Book *books[100];
133         int nbooks;
134         void deleteBook(int pos);
135     public:
136         Library();
137         int Register(Author *a,char *title,int npages);
138         int counthooks(char *a);
139         int lend(char *title);
140         void details();
141         ~Library();
142     };
143
144     Library::Library()
145     {
146         int i;
147         for(i=0;i<100;i++) books[i]=NULL;
148         nbooks=0;
149     }
150
151     void Library::deleteBook(int pos)
152     {
153         int i;
154         delete books[pos];
155         for(i=pos;i<nbooks-1;i++)
156             books[i]=books[i+1];
157         nbooks--;
158     }
159
160     int Library::Register(Author *a,char *title,int npages)
161     {
162         if(nbooks==100) return 0;
163         books[nbooks]=new Book(a,title,npages);
```



Λύση

```
164         nbooks++;
165         return 1;
166     }
167
168     int         Library::countbooks(char *a)
169     {
170         int count=0;
171         int i;
172         for(i=0;i<nbooks;i++)
173         {
174             if(!strcmp(a, books[i]->getTitle()))
175                 count++;
176         }
177         return count;
178     }
179
180     int         Library::lend(char *title)
181     {
182         int i;
183         for(i=0;i<nbooks;i++)
184         {
185             if(!strcmp(books[i]->getTitle(), title))
186             {
187                 deleteBook(i);
188                 return 1;
189             }
190         }
191         return 0;
192     }
193
194     void         Library::details()
195     {
196         int i;
197         for(i=0;i<nbooks;i++)
198         {
199             cout<<"Books: _"<<i<<endl;
200             books[i]->details();
201         }
202     }
203
204     Library::~Library()
205     {
206         int i;
207         for(i=0;i<nbooks;i++) delete books[i];
208     }
209
```



Λύση

```
210 int main()
211 {
212     Author a1("Giannis", "Pappas", 1922, 1998);
213     Author a2("Nikos", "Ioannou", 1967, 0);
214     Library *lib=new Library();
215     lib->Register(&a1, "book1", 100);
216     lib->Register(&a1, "book2", 200);
217     lib->Register(&a1, "book3", 300);
218     lib->details();
219     lib->Register(&a2, "book4", 50);
220     lib->Register(&a2, "book5", 50);
221     lib->Register(&a2, "book6", 100);
222     lib->lend("book2");
223     lib->lend("book5");
224     lib->details();
225     delete lib;
226     return 0;
227 }
```



Άσκηση 1

Να γραφεί κατηγορία με το όνομα **File** για την περιγραφή αρχείων. Στα ιδιωτικά πεδία της κατηγορίας θα περιλαμβάνονται τα ακόλουθα:

1. Όνομα αρχείου
2. Φάκελος του αρχείου (Δείκτης σε *File*)

Στα δημόσιά πεδία

1. Μία μέθοδος δημιουργίας
2. Μέθοδοι *set()* και *get()*
3. Μια αφηρημένη μέθοδος με το όνομα *copy(File *other)* για την αντιγραφή των περιεχομένων του *other* στο τρέχων αρχείο
4. Μια μέθοδος με το όνομα *printPath()* που θα εκτυπώνει την διαδρομή του αρχείου
5. Μια αφηρημένη μέθοδος *details()* που θα εκτυπώνει λεπτομέρειες για το αρχείο.



Άσκηση 1

Να γίνουν δύο νέες κατηγορίες που κληρονομούν την *File* με ονόματα *TextFile* και *Folder*. Στο πρώτο θα υπάρχει στα ιδιωτικά πεδία το αλφαριθμητικό πεδίο *Data* και στο δεύτερο θα υπάρχει ο πίνακας *Contents*, ο οποίος θα είναι τα αρχεία στον φάκελο και θα αποτελείται από δείκτες σε *File*. Κάθε κατηγορία θα υλοποιεί διαφορετικά τις αφηρημένες μεθόδους.



Λύση

```
1 # include <string.h>
2 # include <iostream>
3 using namespace std;
4
5 class File
6 {
7     protected:
8         char *filename;
9         File *parent;
10    public:
11        File(char *f, File *p);
12        const char *getFilename();
13        const File *getParent();
14        virtual void copy(File *other)=0;
15        virtual void details()=0;
16        void printPath();
17        ~File();
18 };
19
20 File::File(char *f, File *p)
21 {
22     filename=new char[ strlen(f)+1];
23     strcpy(filename, f);
24     parent=p;
25 }
26
27 const char *File::getFilename()
28 {
29     return filename;
30 }
31
32
33
34 const File *File::getParent()
35 {
36     return parent;
37 }
38
39 void File::printPath()
40 {
41     if(parent!=NULL) parent->printPath();
42     cout<<" /"<<filename;
43 }
44
45 File::~File()
46 {
```



Λύση

```
47         delete [] filename;
48     }
49
50     class TextFile :public File
51     {
52     private:
53         char *Data;
54     public:
55         TextFile(char *d,char *f,File *p);
56         const char *getData();
57         virtual void copy(File *other);
58         virtual void details();
59         virtual ~TextFile();
60     };
61     TextFile::TextFile(char *d,char *f, File *p):
62         File(f,p)
63     {
64         Data=new char[strlen(d)+1];
65         strcpy(Data,d);
66     }
67
68     const char *TextFile::getData()
69     {
70         return Data;
71     }
72
73     void TextFile::copy(File *other)
74     {
75         delete [] Data;
76         const char *newdata=((TextFile*)other)->getData();
77         Data=new char[strlen(newdata)+1];
78         strcpy(Data,newdata);
79     }
80
81     void TextFile::details()
82     {
83         cout<<"Contents_of_file_"<<filename<<"_are_\n"<<
84             Data<<endl;
85     }
86
87     TextFile::~TextFile()
88     {
89         delete [] Data;
90     }
91
92     class Folder :public File
```



Λύση

```
93 {
94     private:
95         File *Contents[50];
96         int contentsize;
97     public:
98         Folder(char *f, File *p);
99         void addFile(File *f);
100        const int getContentsize();
101        const File *getFileAtPos(int pos);
102        virtual void copy(File *other);
103        virtual void details();
104        virtual ~Folder();
105 };
106
107 Folder::Folder(char *f, File *p)
108     : File(f, p)
109 {
110     contentsize=0;
111 }
112
113 void Folder::addFile( File *f)
114 {
115     Contents[contentsize]=f;
116     contentsize++;
117 }
118
119 void Folder::copy(File *other)
120 {
121     int i;
122     Folder *otherFolder=(Folder *)other;
123     contentsize=((Folder *)other)->getContentsize();
124     for(i=0;i<contentsize;i++)
125     {
126         File *f=(File *)otherFolder->getFileAtPos(i);
127         addFile(f);
128     }
129 }
130 const int Folder::getContentsize()
131 {
132     return contentsize;
133 }
134
135 const File *Folder::getFileAtPos(int pos)
136 {
137     if(pos<0 || pos>=contentsize) return NULL;
138     return Contents[pos];
```



Λύση

```
139 }
140
141 void Folder::details()
142 {
143     int i;
144     cout<<"Folder_"<<filename<<"_files_are_"<<endl;
145     for(i=0;i<contentsize;i++)
146         cout<<Contents[i]->getFilename()<<endl;
147 }
148
149 Folder::~Folder()
150 {
151 }
152
153 int main()
154 {
155     Folder *root=new Folder("/",NULL);
156     Folder *usr=new Folder("usr",root);
157     TextFile *file1=new TextFile("Data_contents_1","filename1",usr);
158     TextFile *file2=new TextFile("Another_data_contents","file2",usr);
159     root->addFile(usr);
160     usr->addFile(file1);
161     usr->addFile(file2);
162     usr->details();
163     return 0;
164 }
```



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 (1/3)



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

