



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

Πρόοδος Αντικειμενοστραφής Προγραμματισμός

Πρόοδος 12/06/2021

ΠΡΟΤΕΙΝΟΜΕΝΑ ΘΕΜΑΤΑ

1. Κατασκευάστε μια κλάση `account` (λογαριασμός τράπεζας) που να περιέχει τα ιδιωτικά δεδομένα: `owner` όνομα ιδιοκτήτη, `balance` υπόλοιπο και `type` τύπος λογαριασμού. Ο τύπος λογαριασμού να είναι `enum class` με όνομα `ACCOUNT_TYPE` και 3 πιθανές τιμές `CHECKING`, `SAVINGS` και `MONEY_MARKET`.
 - Κατασκευάστε `getters` και `σεττερς` για τα ιδιωτικά μέλη δεδομένων της κλάσης αγγουντ.
 - Ορίστε μια στατική μεταβλητή με όνομα `nr_of_accounts` που μετρά πόσα αντικείμενα έχουν κατασκευαστεί.
 - Κατασκευάστε έναν `ζονστρυτορ` που εισάγει τις εξής τιμές: `ιδιοκτήτης=Noname`, `υπόλοιπο=0€`, `τύπος λογαριασμού=CHECKING`
 - Κατασκευάστε έναν `constructor` που να δέχεται 3 παραμέτρους και να αρχικοποιεί με αυτές και τα 3 πεδία του προς κατασκευή αντικειμένου.
 - Κατασκευάστε έναν `copy constructor` που να δημιουργεί ένα αντίγραφο του λογαριασμού αλλά με υπόλοιπο μηδέν.
 - Ορίστε τη συνάρτηση `deposit(double)` (κατάθεση) που προσθέτει στον λογαριασμό ένα ποσό.
 - . Ορίστε τη συνάρτηση `withdrawal(double)` (ανάληψη) που αφαιρεί από τον λογαριασμό ένα ποσό. Αν το ποσό είναι μεγαλύτερο από το υπόλοιπο τότε το υπόλοιπο να μηδενίζεται. Η συνάρτηση να επιστρέφει το ποσό που αφαιρέθηκε από τον λογαριασμό.
 - Υπερφορτώστε τον τελεστή ``` έτσι ώστε να διατάσσει δύο αντικείμενα αγγουντ σύμφωνα με το υπόλοιπό τους.
 - Προσθέστε κατάλληλη συνάρτηση μέλος έτσι ώστε να μετατρέπει ένα αντικείμενο `account` σε `string`.**Κατασκευάστε `main` που :**
 - Δημιουργεί έναν λογαριασμό `acc1` με τον δεφωυλτ ζονστρυτορ.
 - Δημιουργεί έναν λογαριασμό `acc2` με τις τιμές `ιδιοκτήτης=Ken Lay`, `υπόλοιπο=100000€`, `τύπος λογαριασμού=MONEY_MARKET`
 - Δημιουργεί χρησιμοποιώντας τον `copy constructor` έναν τρίτο λογαριασμό `acc3` χρησιμοποιώντας τον λογαριασμό `acc2`.
 - Εμφανίστε το πλήθος των λογαριασμών που έχουν δημιουργηθεί χρησιμοποιώντας την στατική μεταβλητή.
 - Πραγματοποιήστε ανάληψη ενός ποσού που θα δίνει ο χρήστης από τον λογαριασμό `acc2` και κατάθεσή του κατά 80% στο λογαριασμό `acc3` και κατά 20% στο λογαριασμό `acc1`.
 - Εμφανίζει ποιος από τους δύο λογαριασμούς του Ken Lay έχει το μικρότερο υπόλοιπο.

2. Κατασκευάστε μια templated συνάρτηση που να δέχεται ένα διάνυσμα και ένα στοιχείο και να επιστρέφει πόσες φορές υπάρχει το στοιχείο μέσα στο διάνυσμα. Να κληθεί η συνάρτηση :
 - Για ένα διάνυσμα ακεραίων.
 - Για ένα διάνυσμα πραγματικών.
 - Για ένα διάνυσμα με εγγραφές σημείων (struct Point) όπου κάθε σημείο προσδιορίζεται από 2 ακέραιες τιμές.
3. Γράψτε πρόγραμμα που να δημιουργεί έναν πίνακα 5 θέσεων με τις τιμές 2,7,1,3,6. Να δέχεται από τον χρήστη μια ακέραια τιμή x και να αντιγράφει τον πίνακα x φορές σε έναν δυναμικό πίνακα τον οποίο στην συνέχεια να εμφανίζει στην οθόνη.
4. Ποιο το μέγεθος μιας ακέραιας μεταβλητής σε ένα 64bit σύστημα;-**4 byte**
5. Ποιο το μέγεθος ενός δείκτη σε ένα 64 bit σύστημα;-**8 byte**
6. Ποια η λειτουργικότητα του keyword auto;
 - Αυτόματη παραγωγή τιμών
 - Δήλωση σταθεράς μεταβλητής
 - **Αυτόματη απόδοση τύπου δεδομένων**
 - Δέσμευση μίας θέσης μνήμης
7. Τι θα εμφανίσει το ακόλουθο τμήμα κώδικα

```

1  #include <iostream>
2
3  int main()
4  {
5      int *a;
6      *a=10;
7      std::cout<<"A:"<<10<<std::endl;
8      return EXIT_SUCCESS;
9  }
```

- (α') A=10
 - (β') A=0
 - (γ') A=1000
 - (δ') Δεν έχει πραγματοποιηθεί δέσμευση θέσης μνήμης , ούτε ο δείκτης a έχει τιμή.
8. Μία friend συνάρτηση ανήκει στην κλάση;
 - (α') Σωστό
 - (β') Λάθος
 9. Ποιες οι τιμές των X1,X2 στο ακόλουθο τμήμα κώδικα;

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
```

```

4 #include <random>
5 #include <chrono>
6 using namespace std;
7 using namespace std::chrono;
8
9 int main()
10 {
11     auto X1=nullptr;
12     auto X2=nullptr;
13     vector <int> a{1,2,3,4,5,6,7,8,9};
14     shuffle(X1,X2,default_random_engine{steady_clock::now().
time_since_epoch().count()});
15     for(auto &x:a)
16     {
17         cout<<x<<endl;
18     }
19 }

```

- 0,10
- **a.begin(),a.end()**
- a.begin,a.end
- 1,10

10. Δημιουργήστε τις κλάσεις A, B, C και ορίστε από ένα μέλος δεδομένων int σε κάθε μια. Η κλάση να κληρονομεί από την κλάση και να περιέχει ως μέλος δεδομένων ένα αντικείμενο της κλάσης C. Γράψτε συνάρτηση μέλος fun της κλάσης B που να εμφανίζει και τους 3 ακεραίους που περιέχονται σε ένα αντικείμενο της κλάσης. Καλέστε τη συνάρτηση fun από την main.

11. Με ποιόν από τους παρακάτω τρόπους μπορώ να πραγματοποιήσω δυναμική δέσμευση μνήμης για έναν πίνακα 10 δεκαδικών

(α') double b[10];

(β') double *b=new double[10];

(γ') double *b[10];

(δ') double *b=new int(10)

(ε') double *b=(double *)malloc(sizeof(double));

(ς') double *b=(double *)malloc(10)

12. Τι θα εμφανίσει ο ακόλουθος κώδικος;

```

1 #include <iostream>
2
3 using namespace std;
4
5 class Base
6 {
7     protected:
8         int x;
9         int y;
10    public:
11        Base(int x_,int y_):x(x_),y(y_) {}

```

```

12         virtual void avg() const
13         {
14             cout<<"Base Avg:"<<static_cast<double>(this->x+this
->y)/2.0<<endl;
15         }
16     };
17
18     class Derived:public Base
19     {
20     protected:
21         int k;
22     public:
23         Derived(int x_,int y_,int k_):Base(x_,y_),k(k_) {}
24         virtual void avg() const
25         {
26             cout<<"Derived:"<<static_cast<double>(this->x+this->
y+this->k)/3.0<<endl;
27         }
28     };
29
30     class Derived2:public Derived
31     {
32     private:
33         int h;
34     public:
35         Derived2(int x_,int y_,int k_,int h_):Derived(x_,y_,k_),
h(h_) {}
36         void avg() const
37         {
38             cout<<"Derived2:"<<static_cast<double>(this->x+this
->y+this->h+this->k)/4.0<<endl;
39         }
40     };
41
42
43     int main()
44     {
45         Derived2 obj(5,5,5,5);
46         obj.avg();
47     }
48

```

(α') Derived2:5

(β') Base:5

(γ') Derived:5

(δ') Derived2:4

13. Ποιο θα είναι το αποτέλεσμα του παρακάτω κώδικα ;

```

1     #include <iostream>
2
3     int main()
4     {
5         int *a;
6         int k=10;
7         a=&k;

```

```

8      (*a)++;
9      k++;
10     std::cout<<"K="<<k<<std::endl;
11     return EXIT_SUCCESS;
12 }
13

```

(α') k=0

(β') k=12

(γ') k=10

(δ') k=11

14. Έστω η δήλωση δείκτη `int *a=new int(10)`. Με ποιο τρόπο θα απελευθερώσω την μνήμη που δέσμευσα για τον δείκτη.

(α') `free[a]`

(β') `malloc a`

(γ') `delete[] a`

(δ') `delete a`

15. Ποιες από τις παρακάτω μεθόδους αποτελούν μεθόδους ενός vector.

(α') `push_back()`

(β') `remove()`

(γ') `update()`

(δ') `erase()`

(ε') `size`

ΑΣΚΗΣΕΙΣ

- https://github.com/vasnastos/OOP/blob/main/Lesson_6/proodos_1.cpp
- https://github.com/vasnastos/OOP/blob/main/Lesson_6/proodos_2.cpp
- https://github.com/vasnastos/OOP/blob/main/Lesson_6/proodos_3.cpp
- https://github.com/vasnastos/OOP/blob/main/Lesson_6/proodos_4.cpp
- https://github.com/vasnastos/OOP/blob/main/Lesson_6/proodos_5.cpp