

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
Κ10: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ**

Εξετάσεις 21 Σεπτεμβρίου 2017

1. (α') Δώστε μια ομοιότητα και μια διαφορά μεταξύ της C++ και της Java όσον αφορά ορισμό κλάσεων και μια ομοιότητα και μια διαφορά όσον αφορά ορισμό αντικειμένων.
(β') Δίδεται το παρακάτω πρόγραμμα Java:

```
public class MainClass {
    public static void main(String[] args) {
        A a1 = new A(); A a2 = new A(10);
        B b1 = new B(); B b2 = new B(100);
        System.out.println("////////////////////////////////");
        a1.print(); a2.print();
        b1.print(); b2.print();
        System.out.println("////////////////////////////////");
        a1 = b1; a1.print(); } }

class A{
    private int data;
    { System.out.println("A new A was just created"); }
    public A() {
        System.out.println("The A data is " + data); }
    public A(int d) { data = d;
        System.out.println("The A data is " + data); }
    public void print() { System.out.println("The A data is " + data); } }

class B extends A{
    private int data;
    { System.out.println("A new B was just created"); }
    public B() { System.out.println("The B data is " + data); }
    public B(int d) { super(d); data = d;
        System.out.println("The B data is " + data); }
    public void print() { super.print();
        System.out.println("The B data is " + data); } }
```

Επίσης, δίδεται και το παρακάτω πρόγραμμα C++:

```
#include <iostream>
using namespace std;

class A{ int data;
public:
    A(int d = 0) {
        cout << "A new A was just created" << endl;
        data = d;
        cout << "The A data is " << data << endl ; }
    virtual void print() { cout << "The A data is " << data << endl ; } };
```

```

class B : public A{ int data;
public:
    B(int d = 0) : A(d) {
        cout << "A new B was just created" << endl;
        data = d;
        cout << "The B data is " << data << endl ;}
void print() {
    A::print();
    cout << "The B data is " << data << endl; } };

int main() { A a1; A a2(10); B b1; B b2(100);
cout << "////////////////////////////////////" << endl;
a1.print(); a2.print(); b1.print(); b2.print();
cout << "////////////////////////////////////" << endl;
a1 = b1; a1.print(); }

```

Δώστε τα αποτελέσματα του καθενός, αιτιολογώντας την απάντησή σας. Που διαφέρουν και γιατί;

2. Αιτιολογώντας την απάντησή σας, δώστε το αποτέλεσμα της εκτέλεσης του παρακάτω προγράμματος C++:

```

#include<iostream>
using namespace std;

class Block{ int data;
public:
    Block(int datav = 0) : data(datav)
        { cout << "Constructing a Block with " << datav << endl; }
    ~Block(){ cout << "Destructing a Block with: " << data << endl; }
    void double_it() { data = data * 2 ; }
    void print() { cout << data << endl; } };

class B{ Block data1; Block* data2; Block* data3;
public:
    B() : data2(NULL),data3(NULL) { cout << "Constructing a B" << endl; }
    B(int value1, Block* value2) : data1(value1)
        { cout << "Constructing a B with given values " << endl;
          data2 = new Block(value1); data3 = value2; }
    ~B(){ cout << "Destructing a B " << endl;
        if (data2 == NULL) cout << "with NULL data2" << endl;
        else { cout << "with data2: " << endl; delete data2; }
        if (data3 == NULL) cout << "with NULL data3" << endl;
        else{ cout << "with data3: " << endl; data3->print(); }}

    void double_it(){data1.double_it(); data2->double_it(); data3->double_it();}
    void double_data1(){ data1.double_it(); }
    void double_data2(){ data2->double_it(); }
    void double_data3(){ data3->double_it(); } };

int main(){ Block block(20); B b(10,&block);
    block.double_it(); b.double_it(); b.double_data1();
    b.double_data2(); b.double_data3(); block.double_it(); }

```

3. Έστω ότι έχουμε να υλοποιήσουμε σε C++ μία προσομοίωση ενός ταχυφαγείου. Το ταχυφαγείο αποτελείται από ένα σύνολο από “Θέσεις εξυπηρέτησης”. Οι πελάτες που εισέρχονται στο ταχυφαγείο εξυπηρετούνται ένας-ένας από τις “Θέσεις Εξυπηρέτησης” στις οποίες σχηματίζουν ουρές. Όταν εξυπηρετείται ένας πελάτης, δημιουργείται μια παραγγελία. Επίσης, ο πελάτης αφαιρείται από την ουρά της “Θέσης Εξυπηρέτησης” που περίμενε και προστίθεται σε μια λίστα πελατών του ταχυφαγείου σε αναμονή της παράδοσης της παραγγελίας του.

Μια παραγγελία αναφέρεται στον κωδικό του μενού που περιέχει και σχετίζεται με τη “Θέση Εξυπηρέτησης” από την οποία παραγγέλθηκε. Τελικά ανατίθεται σε έναν πελάτη (οποιοδήποτε) που έχει παραγγείλει, από τη “Θέση Εξυπηρέτησης” αυτή, μενού με κωδικό ίδιο με τον κωδικό του μενού της παραγγελίας.

Για την προσομοίωση να χρησιμοποιηθούν οι κλάσεις “Ταχυφαγείο” (`FastFoodRestaurant`), “Θέση εξυπηρέτησης” (`Counter`), “Παραγγελία” (`Order`) και “Πελάτης” (`Customer`).

Η κλάση “Ταχυφαγείο”:

- έχει 5 θέσεις εξυπηρέτησης (`counters`)
- έχει μια λίστα από πελάτες σε αναμονή της παράδοσης της παραγγελίας τους (`waiting_customers`)

Η κλάση “Ταχυφαγείο” χαρακτηρίζεται από την εξής συμπεριφορά:

- Αρχικά η λίστα πελατών σε αναμονή είναι κενή
- Γίνεται άφιξη πελάτη, ο οποίος προστίθεται για να περιμένει στην θέση εξυπηρέτησης στην μικρότερη ουρά (`arrive`)
- Γίνεται εξυπηρέτηση, εξυπηρετώντας πελάτη από τη θέση εξυπηρέτησης με τους περισσότερους πελάτες σε αναμονή, δημιουργώντας μια νέα παραγγελία που αντιστοιχεί στην επιλογή του πελάτη αυτού, αναθέτοντάς του την παραγγελία και προσθέτοντας τον πελάτη στη λίστα πελατών εν αναμονή (`serve`)
- Παραδίδεται παραγγελία σε έναν πελάτη από αυτούς εν αναμονή που ταιριάζει η παραγγελία αφαιρώντας τον από τη λίστα πελατών εν αναμονή (`deliver`)

Η κλάση “Θέση εξυπηρέτησης”:

- έχει μια ουρά από πελάτες (`customer_queue`)
- έχει έναν μετρητή των πελατών που περιμένουν στην ουρά (`waiting_customers`)

Η κλάση “Θέση εξυπηρέτησης” χαρακτηρίζεται από την εξής συμπεριφορά:

- Αρχικά η ουρά των πελατών είναι κενή
- Εξυπηρετεί, αφαιρώντας πελάτη από την ουρά και μειώνοντας τον μετρητή των πελατών που περιμένουν (`serve`)
- Γίνεται άφιξη πελάτη, ο οποίος προστίθεται για να περιμένει στο τέλος της ουράς και αυξάνεται ο μετρητής των πελατών που περιμένουν (`arrive`)

Οι “Παραγγελίες” δομούνται από τον κωδικό του μενού τους και τον αριθμό της “Θέσης εξυπηρέτησης” από την οποία προέκυψαν.

Οι “Πελάτες” δομούνται από τον αριθμό ταυτότητάς τους και την παραγγελία τους. Αρχικά η παραγγελία είναι κενή.

Να υλοποιήσετε σε C++ τις παραπάνω κλάσεις. Δεν χρειάζεται υλοποίηση συνάρτησης `main`.

Σημείωση 1: Όπου θεωρείτε απαραίτητο, κάνετε παραδοχές στις προδιαγραφές, τεκμηριώνοντάς τις.

Σημείωση 2: Θεωρήστε δεδομένο έναν τύπο λίστας κι έναν τύπο ουράς με τις λειτουργίες που χρειάζεστε.