ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

Εργασία Μαθήματος
Πρότυπα Ανάπτυξης Λογισμικού

Αριθμός εργασίας – Τίτλος εργασίας	***Εργασία ISP***
Όνομα φοιτητή	Κωνσταντίνος Καλογερόπουλος
Αρ. Μητρώου	Π19057
Ημερομηνία παράδοσης	22-05-2022

Εκφώνηση εργασίας

Τίτλος:	Εργασία στην αρχή ISP
Περιγραφή:	Αγαπητοί φοιτητές/τριες, για το μάθημα της Πέμπτης 12/5 ως ασύγχρονο online εργαστήριο θα πρέπει να υλοποιήσετε τη **2η εργασία** με βάση τις διαφάνειες 5-Αρχές (βλ. εγγραφα) Α. με βάση την αρχή ISP την προσέγγιση διαχωρισμού μέσω αποστολής μηνυμάτων (delegation) την προσέγγιση της διαφάνειας 63 Β. με βάση την αρχή ISP την προσέγγιση διαχωρισμού με "πολλαπλή" κληρονομικότητα της διαφάνειας 64 **με γλώσσα Java** Δίδεται επίσης έκδοση του ερωτήματος Β σε C++ στα έγγραφα ως υπόδειγμα προσέγγισης **Η εργασία είναι υποχρεωτική, ατομική και πρέπει να παραδοθεί ως 22.5 στην ενότητα Εργασίες του μαθήματος.** Για οποιαδήποτε απορία ή διευκρίνηση επικοινωνήστε στην περιοχή συζητήσεων για την παράδοση. Θα συζητήσουμε περαιτέρω τυχόν απορίες σας στο πλαίσιο του μαθήματος. Ο διδάσκων
Ημερομηνία έναρξης:	05-05-2022 11:26:14
Προθεσμία υποβολής:	22-05-2022 23:59:00 (απομένουν 10 ημέρες 11 ώρες 21 λεπτά)
Τύπος εργασίας:	Ατομική

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1 Εισαγωγή

Στην παρούσα εργασία, γίνεται υλοποίηση των κλάσεων και διεπαφών(interfaces) ενός demo προτζεκτ, ώστε να μην παραβιάζετε η Αρχή Διαχωρισμού των Διασυνδέσεων (Interface Segregation Principle (ISP)). Η αρχή του διαχωρισμού διεπαφής δηλώνει ότι κανένας κώδικας δεν πρέπει να αναγκάζεται να εξαρτάται από μεθόδους που δεν χρησιμοποιεί.

2 Περιγραφή του προγράμματος

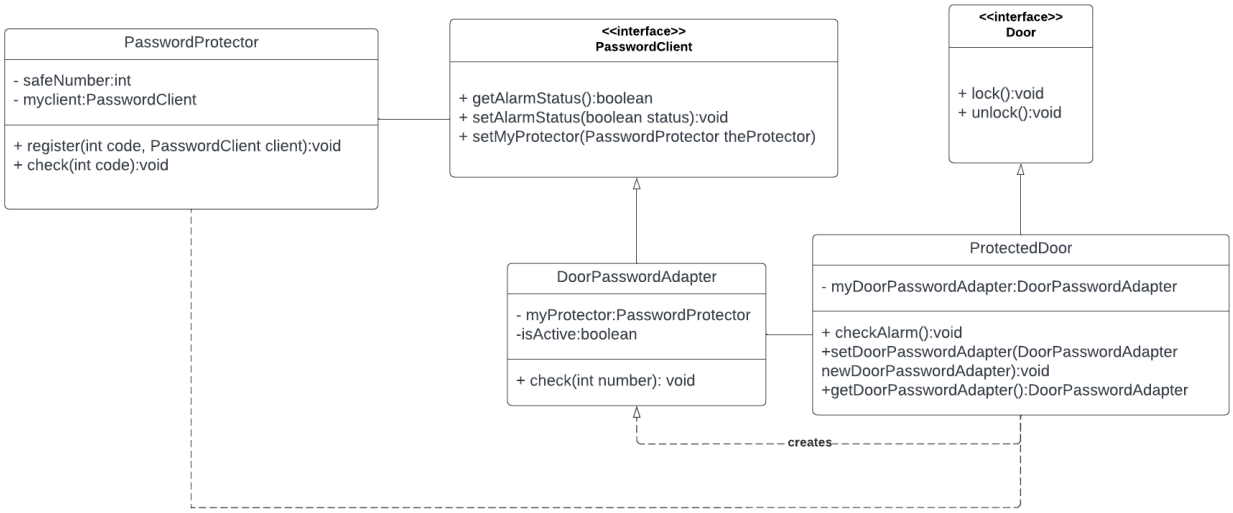
Το πρόγραμμα είναι μια υλοποίηση συναγερμού. Το σύστημα περιέχει μια κλάση ProtectedDoor, η οποία ενεργοποιεί έναν συναγερμό αν παραβιαστεί. Για το σκοπό αυτό, κάθε αντικείμενο ProtectedDoor επικοινωνεί με ένα άλλο αντικείμενο της κλάσης PasswordProtector. Πως θα γίνει η επικοινωνία αυτή, χωρίς περιττή πολυπλοκότητα και εξάρτηση μεθόδων από μια διαφοροποιημένη περίπτωση χρήσης; (Πχ. Υλοποίηση πόρτας χωρίς την χρήση κωδικού)

2.1 Διαχωρισμός μέσω Αποστολής Μηνυμάτων (Delegation) (1^{ος} τροπος)

Αρχικά, δημιουργείται ένα αντικείμενο που υλοποιεί την διεπάφη(Interface) PasswordClient και διαβιβάζει μηνύματα σε ένα αντικείμενο που υλοποιεί την διεπαφή(Interface) Door.

Πιο συγκεκριμένα, η κλάση ProtectedDoor, η οποία ενσωματώνει την διεπαφή μιας πόρτας αλλά και την λειτουργικότητα του κωδικού ασφαλείας, υλοποιεί μόνο την διεπαφή(Interface) Door. Έπειτα, τα αντικείμενα της ProtectedDoor, δημιουργούν αντικείμενου DoorPasswordAdapter, τα οποία υλοποιούν την διεπαφή(Interface) PasswordClient.

Το αντικείμενο DoorPasswordAdapter καταχωρίζεται στο αντικείμενο PasswordProtector και επιπλέον το μήνυμα alarm που αποστέλλει ο PasswordProtector στην περίπτωση παραβίασης, προωθείται στην κλάση ProtectedDoor για τις κατάλληλες ενέργειες.

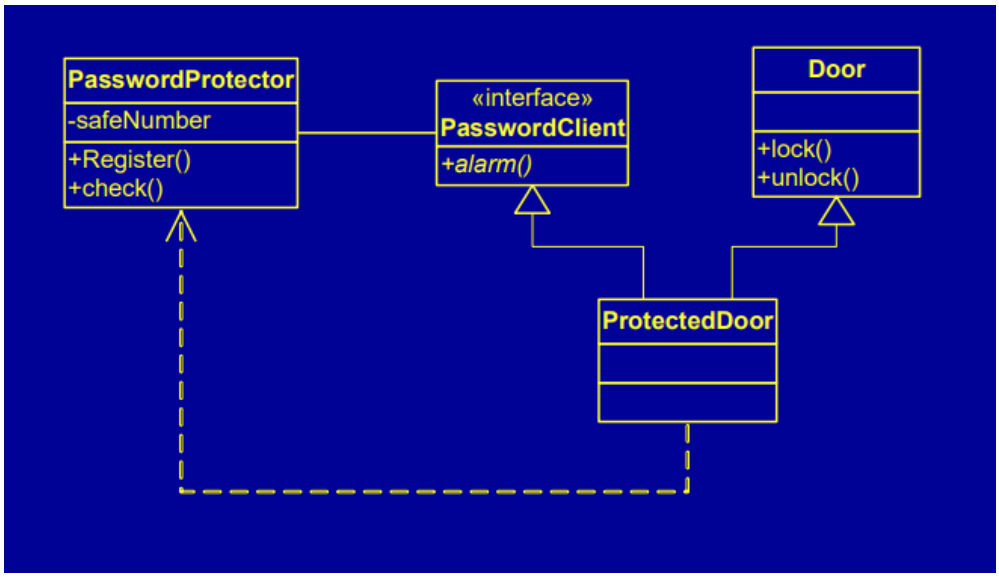


2.2 Διαχωρισμός μέσω Πολλαπλής Κληρονομικότητας (2ος τροπος)

Σε αυτή την περίπτωση, η χρήση πολλαπλής κληρονομικότητας χρήζει το πρόγραμμα πιο εύκολο στην υλοποίηση.

Πιο συγκεκριμένα, η κλάση ProtectedDoor υλοποιεί και τις δυο διεπαφές Door και PasswordClient.

Έτσι, επιτυγχάνεται καλύτερος διαχωρισμός των αρμοδιοτήτων και επιπλέον, οι πελάτες της Door μπορούν να υλοποιήσουν στο μέλλον μια πόρτα χωρίς κωδικό ασφαλείας χωρίς να εξαρτώνται από την ProtectedDoor.



3 Επίδειξη της λύσης

3.1 Παράδειγμα εκτέλεσης του 1ου τρόπου (Delegation).

```
public class App {
    Run | Debug
    public static void main(String[] args) throws Exception {

      PasswordProtector protector = new PasswordProtector(); //creating a new object of PasswordProtector
      ProtectedDoor door = new ProtectedDoor(protector); //creating a new object of Door

      protector.Register(code: 1010, door.getDoorPasswordAdapter()); //registering the door for the first time
      door.lock();
      door.unlock();
      door.checkAlarm();

    }
}
```

1 Παράδειγμα χρήσης του 1ου τρόπου στην μέθοδο main

```
PS C:\Users\kkalo\OneDrive\Documents\GitHub\DesignPatterns\ISP\ISP_multiple\ISP_multiple>  & 'C:
esignPatterns\ISP\ISP_multiple\ISP_multiple\bin' 'App'
The door is locked.
Enter code:
1234
ALARM: Someone is trying to enter without code!!

PS C:\Users\kkalo\OneDrive\Documents\GitHub\DesignPatterns\ISP\ISP_multiple\ISP_multiple>  c:; c
exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kkalo\OneDrive\Documents\GitHub\D
The door is locked.
Enter code:
1010
The door is unlocked.
PS C:\Users\kkalo\OneDrive\Documents\GitHub\DesignPatterns\ISP\ISP_multiple\ISP_multiple>
```

2 Εμφάνιση αποτελεσμάτων στο terminal

3.2 Παράδειγμα εκτέλεσης του 2ου τρόπου (Κληρονομικότητα).

```
public class App {
    Run | Debug
    public static void main(String[] args) throws Exception {

        PasswordProtector myProtector = new PasswordProtector();
        ProtectedDoor mydoor = new ProtectedDoor();

        mydoor.setMyProtector(myProtector); //initializing the protector for the newly created door

        myProtector.Register(code: 1010, mydoor);
        mydoor.lock();

        mydoor.unlock();
    }
}
```

3 Παράδειγμα χρήσης του 2ου τρόπου στην μέθοδο main

```
PS C:\Users\kkalo\OneDrive\Documents\GitHub\DesignPatterns\ISP\ISP_multiple\ISP_multiple>  & 'C:
esignPatterns\ISP\ISP_multiple\ISP_multiple\bin' 'App'
The door is locked.
Enter code:
1234
ALARM: Someone is trying to enter without code!!

PS C:\Users\kkalo\OneDrive\Documents\GitHub\DesignPatterns\ISP\ISP_multiple\ISP_multiple>  c:; c
exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kkalo\OneDrive\Documents\GitHub\D
The door is locked.
Enter code:
1010
The door is unlocked.
PS C:\Users\kkalo\OneDrive\Documents\GitHub\DesignPatterns\ISP\ISP_multiple\ISP_multiple>
```

4 Εμφάνιση αποτελεσμάτων στο terminal

4 Βιβλιογραφικές Πηγές

1. **Αλέξανδρος Ν. Χατζηγεωργίου.** *Αντικειμενοστρεφής Σχεδίαση* : Εκδόσεις Κλειδάριθμος (Σύγγραμα μαθήματος)
2. Διαφάνειες μαθήματος στο e-class (5-Αρχές)