



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

**ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ - ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ
ΚΑΙ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ**

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ (JAVA)

ΑΝΑΦΟΡΑ ΠΡΩΤΗΣ ΑΤΟΜΙΚΗΣ ΕΡΓΑΣΙΑΣ

ΜΕΤΑΠΤΥΧΙΑΚΟΣ ΦΟΙΤΗΤΗΣ:

- **ΝΙΚΟΛΑΣ ΠΑΤΕΡΑΣ – ΜΠΣΠ21043**

Κύπρος, Ιανουάριος 2022

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	2
ΠΙΝΑΚΑΣ ΕΙΚΟΝΟΓΡΑΦΙΩΝ	3
1. ΠΡΟΤΥΠΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	4
1.1. ABSTRACT FACTORY	4
1.2. OBSERVER.....	5
1.3. SINGLETON	5
2. ΕΙΚΟΝΟΓΡΑΦΙΕΣ.....	6
3. ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΠΗΓΕΣ	8
4. ΒΙΒΛΙΟΘΗΚΕΣ & ΕΡΓΑΛΕΙΑ	9

ΠΙΝΑΚΑΣ ΕΙΚΟΝΟΓΡΑΦΙΩΝ

Εικόνα 2.1: Ζητούμενα από τον χρήστη.....	6
Εικόνα 2.2: Όταν κατασκευαστεί κινητό και ενεργοποιηθεί ο observer.	6
Εικόνα 2.3: Όταν ο πελάτης παραλάβει ένα Featurephone.	6
Εικόνα 2.4: Πελάτες που παίρνουν κινητό.....	7
Εικόνα 2.5: Παράδειγμα 1	7
Εικόνα 2.6: Παράδειγμα 2	7

1. ΠΡΟΤΥΠΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

1.1. ABSTRACT FACTORY

ΥΛΟΠΟΙΗΣΗ:

Το σχεδιαστικό πρότυπο «Abstract Factory» δημιουργεί ένα interface για τη δημιουργία διάφορων οικογενειών κλάσεων. Στην παρούσα εργασία χρησιμοποιήθηκε για την δημιουργία ενός interface το οποίο μπορεί να επιστρέψει κάποια χαρακτηριστικά του τηλεφώνου που δημιουργείται σαν τύπο, μέγεθος οθόνης και χωρητικότητα.

Η διεπαφή αυτή δηλώνει ένα σύνολο μεθόδων δημιουργίας που μπορεί να χρησιμοποιηθεί για την παραγωγή διαφορετικών τύπων στοιχείων διεπαφής τηλεφώνου. Τα «concrete» εργοστάσια αντιστοιχούν σε συγκεκριμένους τύπους τηλεφώνων (smartphones & featurephones) και δημιουργούν τα στοιχεία διεπαφής τηλεφώνου που ταιριάζουν με το συγκεκριμένο τηλέφωνο. Το πρόγραμμα χρησιμοποιεί αυτές τις πληροφορίες για να δημιουργήσει ένα εργοστασιακό αντικείμενο από μια κλάση που ταιριάζει με το τηλέφωνο.

Ο υπόλοιπος κώδικας χρησιμοποιεί αυτό το εργοστάσιο για τη δημιουργία στοιχείων διεπαφής. Με αυτήν την προσέγγιση, ο κώδικας δεν εξαρτάται από συγκεκριμένες κατηγορίες εργοστασίων και στοιχείων διεπαφής χρήστη, εφόσον λειτουργεί με αυτά τα αντικείμενα μέσω των αφηρημένων διεπαφών τους. Αυτό επιτρέπει επίσης στον κώδικα να υποστηρίζει άλλα εργοστάσια ή στοιχεία διεπαφής που ενδέχεται να προστεθούν στο μέλλον. Ως αποτέλεσμα, δεν χρειάζεται να τροποποιείται ο κώδικας κάθε φορά που προσθέτετε μια νέα παραλλαγή στοιχείων διεπαφής. Απλώς πρέπει να δημιουργηθεί μια νέα εργοστασιακή κλάση που παράγει αυτά τα στοιχεία και να τροποποιηθεί ελαφρώς ο κώδικας προετοιμασίας του προγράμματος, ώστε να επιλέγει αυτήν την κλάση όταν χρειάζεται.

1.2. OBSERVER

ΥΛΟΠΟΙΗΣΗ:

Ο «Observer» είναι ένα πρότυπο σχεδιασμού συμπεριφοράς. Καθορίζει την επικοινωνία μεταξύ αντικειμένων: observable και observers. Ένα observable είναι ένα αντικείμενο που ειδοποιεί τους observers για τις αλλαγές στην κατάστασή του.

Ειδικότερα, χρησιμοποιείται για να ενημερώσουμε τους πελάτες όταν κατασκευαστεί ένα καινούργιο κινητό. Υπάρχει ένας observable που είναι ένα αντικείμενο «Staff» δηλαδή προσωπικό και θα ειδοποιεί τους observers που είναι οι πελάτες. Επίσης, δημιουργήθηκαν διαφορετικά interfaces για κάθε τύπο κινητού (Smartphone & Featurephone).

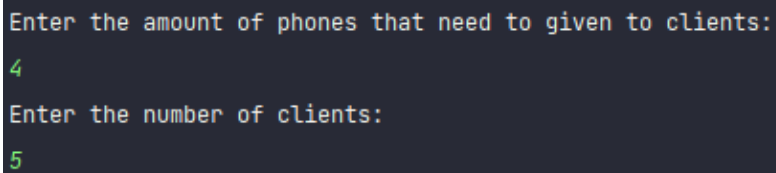
1.3. SINGLETON

ΥΛΟΠΟΙΗΣΗ:

Το πρότυπο «Singleton» λέει ότι απλώς “ορίστε μια κλάση που έχει μόνο ένα instance”. Χρησιμοποιήθηκε η προσέγγιση με «InstanceHolder» η οποία μας δίνει τα εξής πλεονεκτήματα:

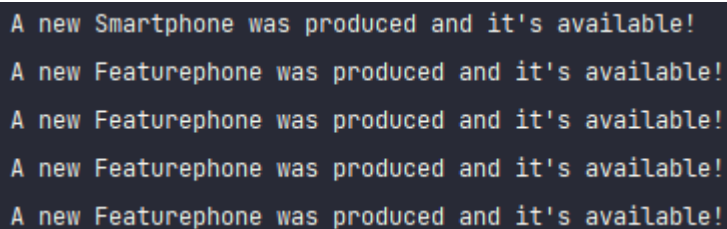
- Ασφάλεια από νήματα.
- Άκρως ταυτόχρονη lazy προετοιμασία στατικών πεδίων με καλή απόδοση.

2. ΕΙΚΟΝΟΓΡΑΦΙΕΣ



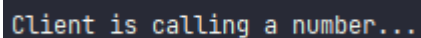
```
Enter the amount of phones that need to given to clients:
4
Enter the number of clients:
5
```

Εικόνα 2.1: Ζητούμενα από τον χρήστη.



```
A new Smartphone was produced and it's available!
A new Featurephone was produced and it's available!
A new Featurephone was produced and it's available!
A new Featurephone was produced and it's available!
A new Featurephone was produced and it's available!
```

Εικόνα 2.2: Όταν κατασκευαστεί κινητό και ενεργοποιηθεί ο observer.



```
Client is calling a number...
```

Εικόνα 2.3: Όταν ο πελάτης παραλάβει ένα Featurephone.

```
Enter the amount of phones that need to given to clients:
4
Enter the number of clients:
5
A new Smartphone was produced and it's available!
A new Smartphone was produced and it's available!
A new Featurephone was produced and it's available!
A new Featurephone was produced and it's available!
A new Featurephone was produced and it's available!

Client is taking a picture...
Client is calling a number...
Client is taking a picture...
```

Εικόνα 2.4: Πελάτες που παίρνουν κινητό.

Παραδείγματα Εκτέλεσης:

```
Enter the amount of phones that need to given to clients:
4
Enter the number of clients:
5
A new Smartphone was produced and it's available!
A new Smartphone was produced and it's available!
A new Featurephone was produced and it's available!
A new Featurephone was produced and it's available!
A new Featurephone was produced and it's available!

Client is taking a picture...
Client is calling a number...
Client is calling a number...
Client is taking a picture...

Clients without phone: 1
```

Εικόνα 2.6: Παράδειγμα 2

```
Enter the amount of phones that need to given to clients:
4
Enter the number of clients:
5
A new Featurephone was produced and it's available!
A new Featurephone was produced and it's available!

Client is calling a number...
Client is calling a number...

Clients without phone: 3
```

Εικόνα 2.5: Παράδειγμα 1

3. ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΠΗΓΕΣ

1. Παραδείγματα κώδικα που δόθηκαν στην τάξη.
2. <https://www.javatpoint.com/singleton-design-pattern-in-java>
3. <https://refactoring.guru/design-patterns/observer/java/example#:~:text=Observer%20is%20a%20behavioral%20design,that%20implements%20a%20subscriber%20interface.>
4. <https://www.baeldung.com/java-observer-pattern>
5. https://www.tutorialspoint.com/design_pattern/observer_pattern.htm
6. <https://medium.com/@kevalpatel2106/how-to-make-the-perfect-singleton-de6b951dfdb0#:~:text=1.,want%20to%20create%20the%20Singleton.>

4. ΒΙΒΛΙΟΘΗΚΕΣ & ΕΡΓΑΛΕΙΑ

OpenJDK-17 Version 17.0.1
IntelliJ
java.util.*
java.util.List
java.util.ArrayList
import java.util.Objects
import java.util.Random