

# Εργασία 3

Ημερομηνία παράδοσης 5 Ιουνίου 2021

---

Εργαστήριο: Τεχνολογία Λογισμικού

Τρίτη εργασία εαρινού εξαμήνου 2020-2021



# Επισκόπηση

---

Η εργασία έχει στόχο την δημιουργία μιας απλής εφαρμογής για την διαχείριση αποθήκης ενός καταστήματος ειδών πληροφορικής.

Η εφαρμογή θα πρέπει να υποστηρίζει τις παρακάτω κεντρικές λειτουργίες:

1. Αναζήτηση είδους από Αποθήκη
2. Παραγγελία ειδών,
3. Τιμολόγηση



# Επισκόπηση συνέχεια

---

Η συγκεκριμένη αποθήκη περιέχει:

- είδη πληροφορικής (κλάση Inventory),
- έχει πελάτες (κλάση Customer),
- δέχεται παραγγελίες (κλάση Order) και
- εκδίδει τιμολόγια (μόνο εκτύπωση)

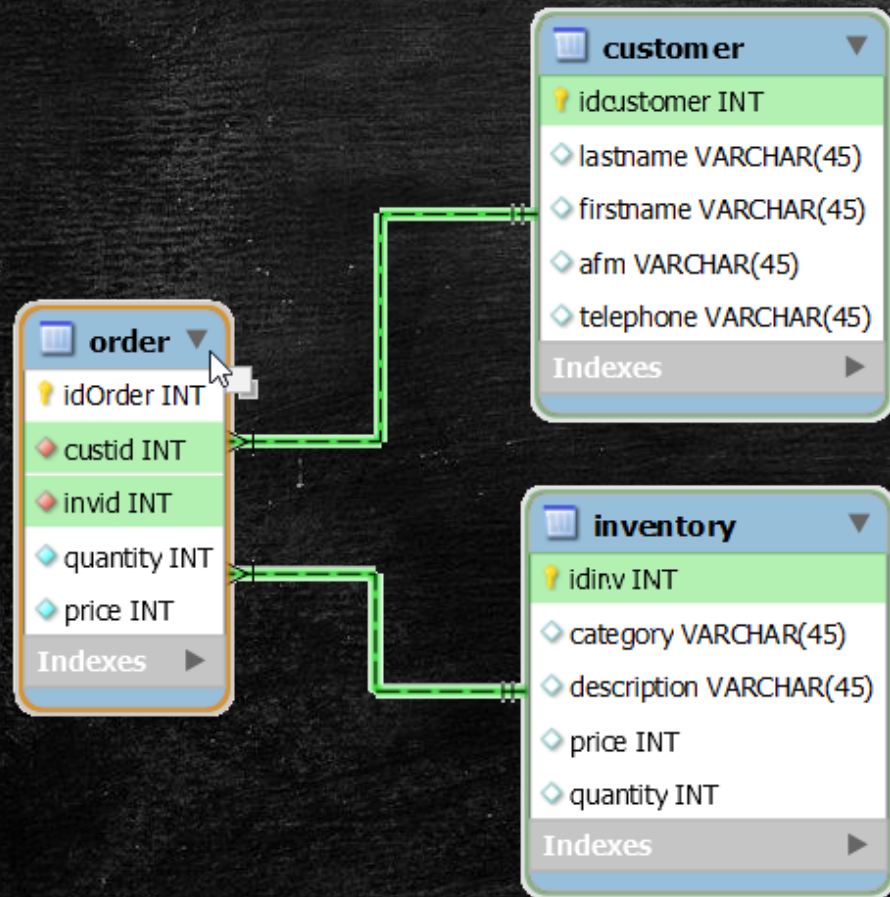


# Λειτουργικές απαιτήσεις

- Η εφαρμογή θα βασίζεται σε ένα κεντρικό μενού, το οποίο θα οδηγεί σε φόρμες, σαν αυτή που κάνατε στην πρόοδο.
- ΟΛΕΣ οι φόρμες που θα φτιάξετε, θα πρέπει να επιτρέπουν την εισαγωγή, παρουσίαση, μετατροπή, αναζήτηση και διαγραφή πληροφορίας.
- Όλα οι πληροφορίες θα αποθηκεύονται σε μια σχεσιακή βάση δεδομένων.
- Η φόρμα έκδοσης τιμολογίου θα πρέπει:
  - να συνδυάζει τα στοιχεία του πελάτη και τα στοιχεία των ειδών,
  - να ζητάει ποσότητα,
  - να υπολογίζει συνολικό κόστος γραμμής,
  - να υπολογίζει συνολικό κόστος τιμολογίου και
  - να τυπώνει το αποτέλεσμα με `System.out.println()`.



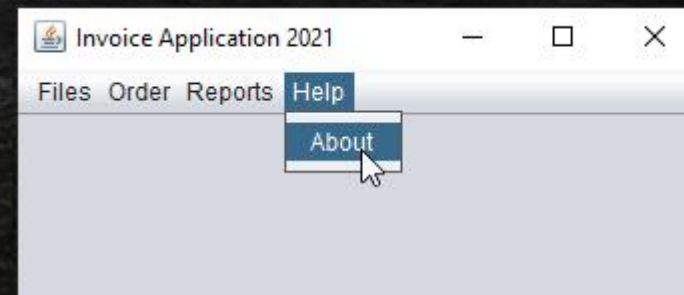
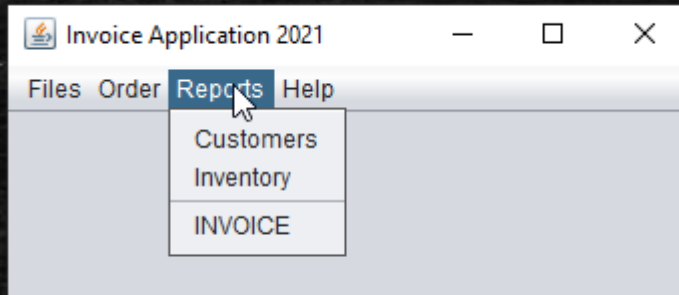
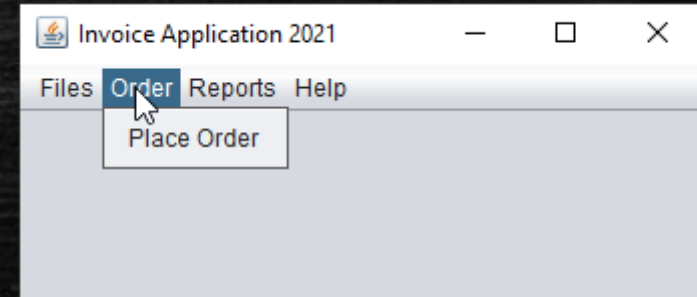
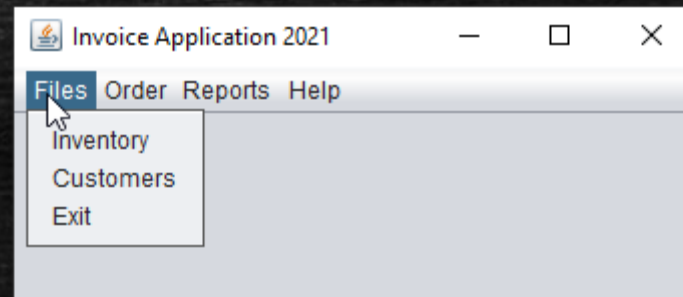
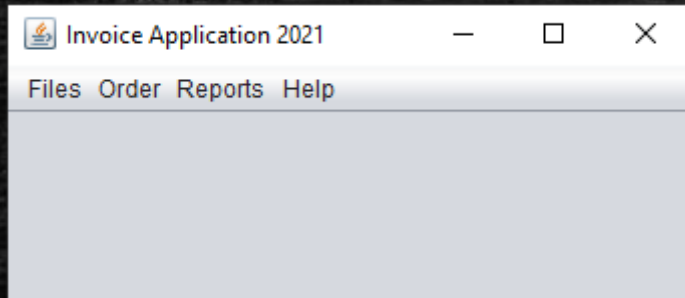
# Entity Relational Diagram της Βάσης Δεδομένων



- Πίνακες: order, customer και inventory
- Foreign keys: custid και invid
- Στις σημειώσεις εργαστηρίου θα βρείτε το sql script που κατασκευάζει το σχήμα
- Το τιμολόγιο θα «τυπωθεί» χρησιμοποιώντας τα στοιχεία από το order σε συνδυασμό με τους άλλους 2 πίνακες



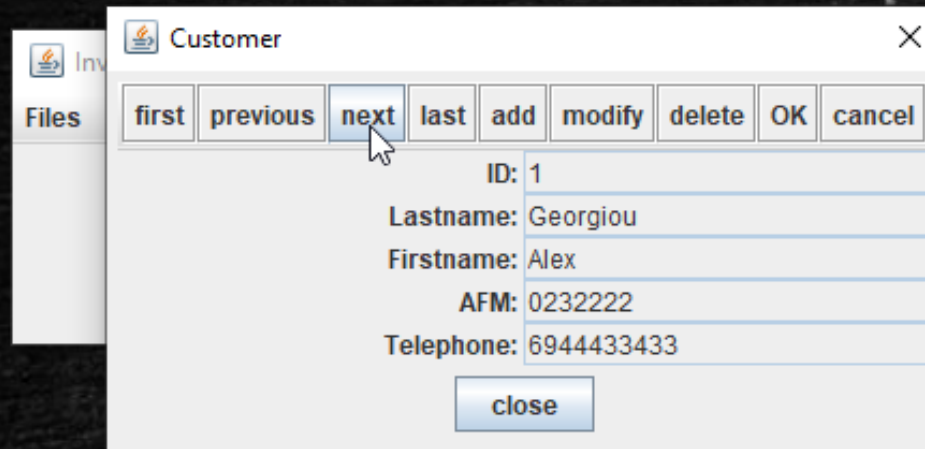
# Τα μενού επιλογών που πρέπει να υλοποιηθούν





# Διάλογοι

- Τα JMenuItemς πρέπει να έχουν ActionListeners που να ανοίγουν modally διάλογους.
- Δείγμα κώδικα που μπορείτε να χρησιμοποιήσετε σαν βάση για την υλοποίηση σας βρίσκεται στο eclass: Εργαστήριο 09, 13. **Formz4Data**.
- Παράδειγμα υλοποίησης της φόρμας Customer:





# Εισαγωγή Παραγγελίας

- Για να γίνεται εύκολα η παραγγελία καλό είναι να γίνει χρήση JComboBox
- Τα ComboBoxes χωρίς πληκτρολόγηση μπορούν να δείξουν τα περιεχόμενα της βάσης δεδομένων
- Για απλούστευση των διαδικασιών θα βάλουμε την ποσότητα σταθερά 1, οπότε η τιμή του αντικειμένου θα είναι και η τελική τιμή σε κάθε γραμμή. Αν θέλει κάποιος να παραγγείλει 2 πληκτρολόγια πχ, θα πρέπει να βάλει 2 παραγγελίες από 1.
- Επιπλέον, για λόγο απολοποίησης, όπως ίσως παρατηρήσατε, στη βάση δεν κρατάμε ημερομηνία παραγγελίας, οπότε κάθε πελάτης θα δημιουργεί μία παραγγελία με πολλά είδη και θα δέχεται ένα τιμολόγιο.
- Καλό θα ήταν να αλλάζατε το όνομα το πίνακα order σε orders. Αλλιώς θα πρέπει να τον ζητάτε σαν inv.order επειδή η λέξη order είναι δεσμευμένη λέξη στην SQL.



# Διάλογος Παραγγελίας

- Ο διάλογος παραγγελίας είναι το σημείο στο οποίο συνδυάζονται οι πελάτες με τα προϊόντα.
- Για την υλοποίηση χρησιμοποιούμε JSplitPane με VERTICAL\_SPLIT και χωρίζουμε το ContentPane σε δύο κομμάτια, έχοντας ταυτόχρονα έλεγχο του μεγέθους τους.

Place Order

Customer: Georgiou, Alex, 1

Inventory Item: Laptop, Dell Vostro I5 , 400, 1

Item Price: 400

Quantity: 1

Total Price: 400

Add Line Delete Line

| OrderID | Customer  | Category | Description    | Price |
|---------|-----------|----------|----------------|-------|
| 1       | Georgiou  | SSD      | Kingston 480G  | 60    |
| 2       | Georgiou  | Laptop   | Dell Vostro I5 | 400   |
| 4       | Dimitriou | Laptop   | Dell Vostro I5 | 400   |
| 5       | Georgiou  | HDD      | Seagate 4TB    | 100   |

Exit

1

Place Order

Customer: Georgiou, Alex, 1

Inventory Item: Laptop, Dell Vostro I5 , 400, 1

Item Price: Laptop, Dell Vostro I5 , 400, 1  
Laptop, HP 15S Ryzen, 500, 2  
Keyboards, Logitech, 65, 3  
Total Price: HDD, Seagate 4TB, 100, 4  
SSD, Kingston 480G, 60, 5  
SSD NVME, Samsung 500G, 90, 6

Quantity: 1

Add Line Delete Line

| OrderID | Customer  | Category | Description    | Price |
|---------|-----------|----------|----------------|-------|
| 1       | Georgiou  | SSD      | Kingston 480G  | 60    |
| 2       | Georgiou  | Laptop   | Dell Vostro I5 | 400   |
| 4       | Dimitriou | Laptop   | Dell Vostro I5 | 400   |
| 5       | Georgiou  | HDD      | Seagate 4TB    | 100   |

Exit

2

Place Order

Customer: Georgiou, Alex, 1

Inventory Item: Laptop, Dell Vostro I5 , 400, 1

Item Price: 400

Quantity: 1

Total Price: 400

Add Line Delete Line

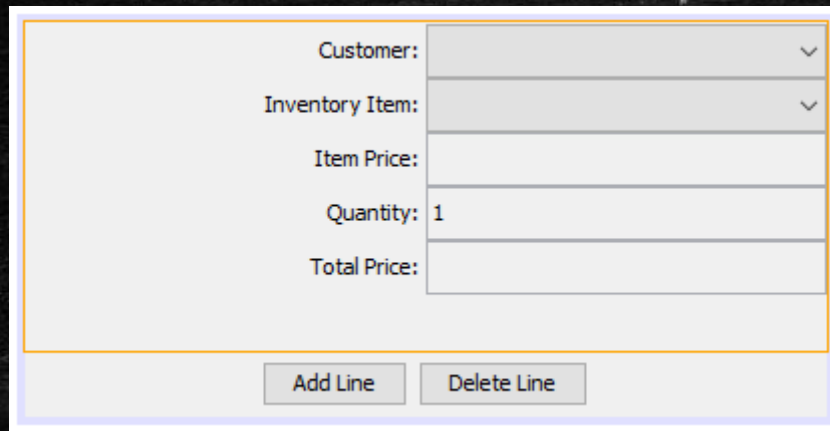
| OrderID | Customer  | Category | Description    | Price |
|---------|-----------|----------|----------------|-------|
| 1       | Georgiou  | SSD      | Kingston 480G  | 60    |
| 2       | Georgiou  | Laptop   | Dell Vostro I5 | 400   |
| 4       | Dimitriou | Laptop   | Dell Vostro I5 | 400   |
| 5       | Georgiou  | HDD      | Seagate 4TB    | 100   |
| 6       | Georgiou  | SSD NVME | Samsung 500G   | 90    |

Exit



# Το επάνω JPanel της Παραγγελίας

- Έχει BorderLayout:
- Στο κέντρο φιλοξενεί μία φόρμα με GridLayout 6 γραμμές 2 στήλες με ζεύγη JLabel και JComboBox ή JTextField. Η έκτη γραμμή έχει δύο JLabels χωρίς λεζάντα για να δημιουργήσει χώρο. Το Quantity είναι πάντα 1, οπότε το Total Price είναι πάντα ίδιο με το Item Price.
- Στο South έχει ένα JPanel με FlowLayout με δύο κουμπιά.



The screenshot shows a Java Swing window with a light gray background. It contains a form with five rows of input fields, each preceded by a label. The first two rows are for 'Customer' and 'Inventory Item', both using JComboBox (dropdown menus). The next three rows are for 'Item Price', 'Quantity' (with a value of 1), and 'Total Price', all using JTextField. Below the form, there are two buttons: 'Add Line' and 'Delete Line'. The entire form is enclosed in a white border with a thin orange border around the input fields.

|                 |                      |
|-----------------|----------------------|
| Customer:       | <input type="text"/> |
| Inventory Item: | <input type="text"/> |
| Item Price:     | <input type="text"/> |
| Quantity:       | 1                    |
| Total Price:    | <input type="text"/> |

Add Line Delete Line



# Τα κουμπιά από το πάνω Panel

- Όταν πατηθεί το Add Line, ο Listener μαζεύει τις πληροφορίες από τα 2 JComboBoxes και την ποσότητα και γράφει στη βάση, στον πίνακα orders. Μετά «φρεσκάρει» το JTable του κάτω JPanel, έτσι ώστε να φαίνεται η νέα γραμμή.
- Το Delete Line δουλεύει σε συνδυασμό με το JTable. Κλικάρουμε στη γραμμή που επιθυμούμε να διαγράψουμε και διαγράφει την γραμμή και ενημερώνει το JTable.

Place Order

Customer: Georgiou, Alex, 1

Inventory Item: Laptop, Dell Vostro I5 , 400, 1

Item Price: 400

Quantity: 1

Total Price: 400

Add Line Delete Line

| OrderID | Customer  | Category | Description    | Price |
|---------|-----------|----------|----------------|-------|
| 1       | Georgiou  | SSD      | Kingston 480G  | 60    |
| 2       | Georgiou  | Laptop   | Dell Vostro I5 | 400   |
| 4       | Dimitriou | Laptop   | Dell Vostro I5 | 400   |
| 5       | Georgiou  | HDD      | Seagate 4TB    | 100   |
| 6       | Georgiou  | SSD NVME | Samsung 500G   | 90    |

Exit

Delete

Place Order

Customer: Georgiou, Alex, 1

Inventory Item: Laptop, Dell Vostro I5 , 400, 1

Item Price: 400

Quantity: 1

Total Price: 400

Add Line Delete Line

| OrderID | Customer  | Category | Description    | Price |
|---------|-----------|----------|----------------|-------|
| 1       | Georgiou  | SSD      | Kingston 480G  | 60    |
| 2       | Georgiou  | Laptop   | Dell Vostro I5 | 400   |
| 4       | Dimitriou | Laptop   | Dell Vostro I5 | 400   |
| 6       | Georgiou  | SSD NVME | Samsung 500G   | 90    |

Exit



# Το κάτω JPanel της παραγγελίας

- Έχει BorderLayout:
- Στο κέντρο φιλοξενεί ένα JScrollPane με ένα JTable που περιέχει τα στοιχεία των γραμμών όλων των παραγγελιών.
- Στο South έχει ένα JToolBar με ένα κουμπί “Exit” που κλείνει τον διάλογο.

| OrderID | Customer  | Category | Description    | Price |
|---------|-----------|----------|----------------|-------|
| 1       | Georgiou  | SSD      | Kingston 480G  | 60    |
| 2       | Georgiou  | Laptop   | Dell Vostro I5 | 400   |
| 4       | Dimitriou | Laptop   | Dell Vostro I5 | 400   |
| 5       | Georgiou  | HDD      | Seagate 4TB    | 100   |

Exit



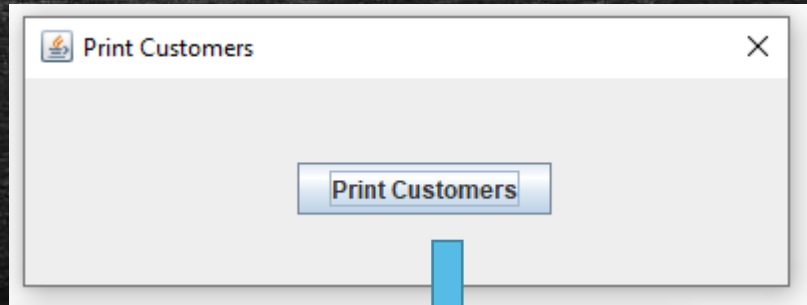
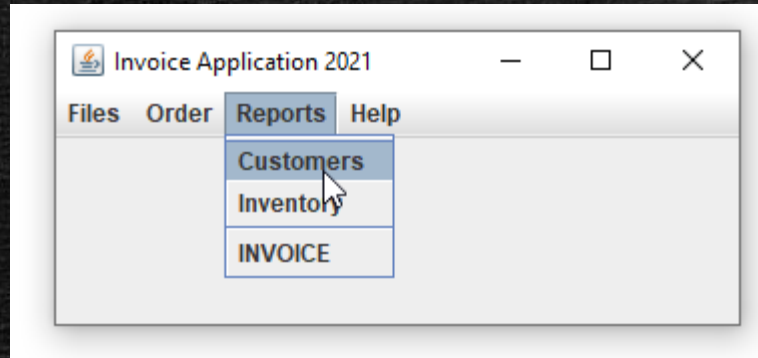
# SQL Statement

- Χρησιμοποιούμε foreign keys για να αποφεύγουμε το Redundancy. Ο πίνακας orders δεν έχει λεπτομέρειες για τον κάθε πελάτη, ούτε για το κάθε προϊόν επειδή περιέχει τα κλειδιά τους.
- Για να τροφοδοτηθεί το JTable χρησιμοποιούμε το παρακάτω SQL command:

```
SELECT *  
  FROM orders  
    left join inv.customer on custid=idcustomer  
    left join inv.inventory on invid=idinv;
```



# Customers Report



The screenshot shows the 'Output' tab of a test results window. It displays a table titled 'CUSTOMER REPORT' with the following data:

| Code | Lastname  | Firstname | AFM     | Telephone |
|------|-----------|-----------|---------|-----------|
| 1    | Georgiou  | Alex      | 0232222 | 694443343 |
| 2    | Stavrakis | Petros    | 2001233 | 674135023 |
| 3    | Dimitriou | Nikos     | 4221222 | 674243220 |



# Εκτύπωση Invoice

The sequence of screenshots illustrates the workflow for printing an invoice:

- Invoice Application 2021**: The user navigates to the **Reports** menu and selects **INVOICE**.
- Print Invoice**: The application prompts for the customer. The initial selection is **Georgiou, Alex, 1**.
- Print Invoice**: The dropdown menu is expanded, showing a list of customers: **Dimitriou, Nikos, 3**, **Georgiou, Alex, 1**, **Stavrakis, Petros, 2**, and **Dimitriou, Nikos, 3**.
- Print Invoice**: The customer **Dimitriou, Nikos, 3** is selected.

The final screenshot shows the output of the test layout for Customer ID 3:

```
Notifications | Output x | Search Results
TestLayout (run-single) x | TestLayout (run-single) #2 x
For Customer ID: 3
Customer Name: Dimitriou Nikos
=====
Order Category      Description      Quantity  Price
=====
4 Laptop            Dell Vostro I5   1         400
=====
```



# Εργαστήρια Ερχόμενης Τρίτης

---

- Την Τρίτη 25 Μαΐου 2021, θα μιλήσουμε για:
- Generics
- JComboBox και ComboBoxModel
- Jtable και AbstractTableModel
- Prepared Statements στο JDBC



# ΚΑΛΗ ΕΠΙΤΥΧΙΑ!

---

Περιγράφονται οι ελάχιστες απαιτήσεις σε λειτουργικότητα και εμφάνιση