Софийски Университет "Климент Охридски"
Факултет по Математика и Информатика

Финален изпит No. 1а

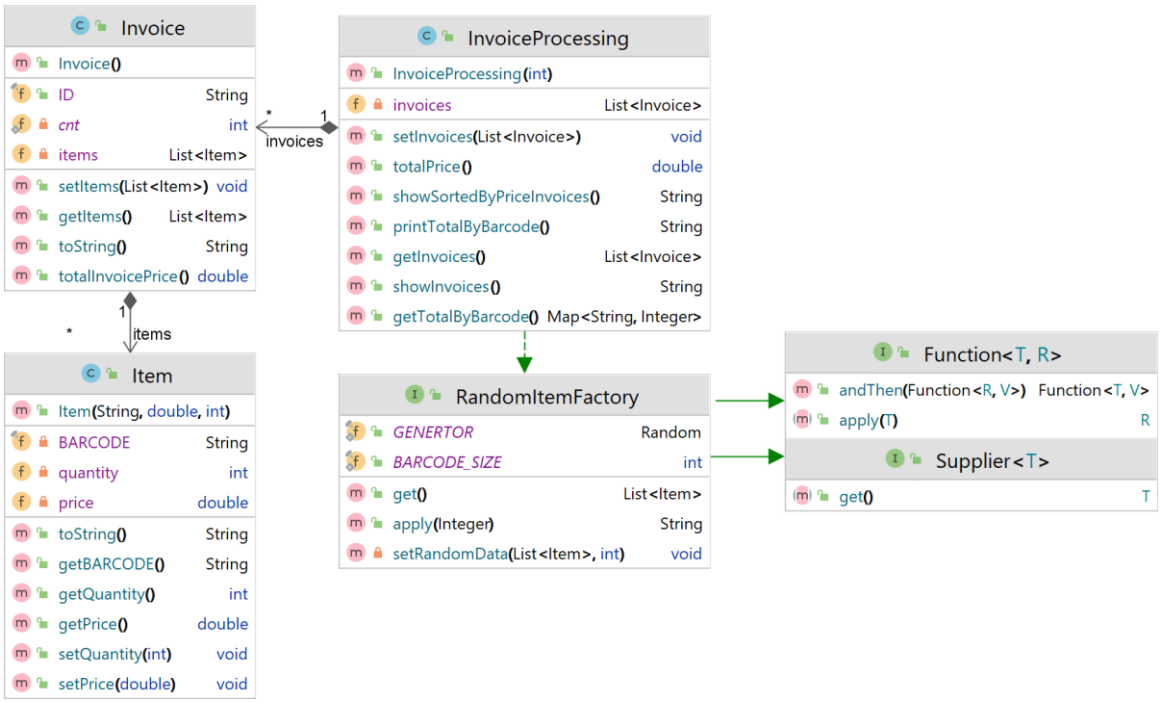
Курс: **Приложно Обектно Ориентирано Програмиране с Java, част 1**
Преподавател: проф. д-р. Е. Кръстев **Студент** :
Дата: юни 2022 **Време за работа: 120 min**

Инструкции: **Изпълнете следното задание и предайте в своя акаунт в Мудъл пълния набор от файлове на IntelliJ проекта, създаден за решаване на програмата. Пълен набор от точки се присъжда за пълно и коректно решение на всички подзадачи.**
Всеки студент носи отговорност за пълнотата на предаденото решение. Каквото е предадено, само това се оценява
Оценки:

- **2 от 0 до 54 точки**
- **3 от 55 до 64 точки**
- **4 от 65 до 74 точки**
- **5 от 75 до 84 точки**
- **6 от 85 до 100 точки**

Задача 1 (100 точки)

Задание за програмиране.



A. Създайте проект на IntelliJ с **Java модул** именуван като **`exam.logic`**, съответен на модула **`package exam.logic`** и файл **`module-info.java`** с описание на модула.. **Създайте в пакета `exam.logic` следните типове данни (показани в UMLклас диаграми по-горе за яснота)-**

Точки: 4

1. **Клас `Item`, който има `BARCODE`, нестатична константа от тип `String`, а също `quantity` и `price` съответно от тип `int` `double`. Инициализирайте тези данни в конструктор за общо ползване. Добавете коректна дефиниция на `Getter` и `Setter`**

за тези данни. Предефинирайте метода `toString()` да извежда във форматиран вид всички данни на `Item`, както е показано в примерното изпълнение в края на текста.

Точки: 8

2. Клас `Invoice`, който има уникален идентификатор `ID`, нестатична константа от тип `String`. Добавете също данна `items` от тип `List<Item>`. Инициализирайте тези данни в конструктор по подразбиране, където стойността на `ID` е съставена от `7` цифри, а незначещите цифри са нули. Добавете коректна дефиниция на `Getter` и `Setter` за данната `items`. Предефинирайте метода `toString()` да извежда форматиран `String` с всички данни на `Invoice`, с `Item` данните на отделни редове и сортирани възходящо по `BARCODE`, както е показано в примерното изпълнение в края на текста.

   Добавете към този клас метод

   `double totalInvoicePrice( )`

   който връща общата стойност на фактурата `Invoice` (суми от цена по количество)

   Точки: 12

3. Напишете интерфейса `RandomItemFactory`, който наследява функционалните интерфейси `Function<Integer, String>>` и `Supplier<List<Item>>`. Добавете в този интерфейс следните

   Данни:

   - Константа `GENERATOR` от тип `Random`, инициализирана по подразбиране
   - Константа `BARCODE_SIZE` от тип `int`, инициализирана по подразбиране на 5

   методи:

   a) Метод по подразбиране

   `String apply(Integer num)`

   който генерира баркод, съставен от `num` на брой произволно генерирани символи `':'` и `'|'` посредством данната `GENERATOR`

   b) `private` метод

   `void setRandomData(List<Item> items, int count)`

   който добавя обекти в списъка `items`, при което данните `BARCODE`, `quantity` и `price` се генерират съответно с `apply(BARCODE_SIZE)` и `GENERTOR` (`quantity ∈ [1,10]`, `price ∈ [20,30]`)

   c) Метод по подразбиране

   `List<Item> get()`

   който създава `List<Item>` и използва метода `setRandomData` за добавяне към този списък на 1 до 6 обекта `Item` (броят обекти да се избере с `GENERATOR`)

   Точки: 16

4. Напишете `class InvoiceProcessing`, чиито обекти има данна `List<Invoice> invoices` и имплементирайте интерфейса `RandomItemFactory` в този клас. Инициализирайте `invoices` в конструктор на `InvoiceProcessing` с целочислен параметър, задаващ броя на елементите в `invoices`. Данната `items` на

тези елементи инициализирайте с помощта на метода `get()`. Добавете коректна дефиниция на `Getter` и `Setter` за данната `invoices`.

Точки: 6

5. **Напишете** следните методи в **`class InvoiceProcessing`**

   a) **Метод**

   `public String showInvoices()`

   Използва задължително Stream API за създаване на `String`, където елементите на `invoices` се изписват един след друг на отделни редове като се използва метода `toString()` на `Invoice`

   (4 точки)

   b) **Метод**

   `public double totalPrice()`

   Използва задължително Stream API за пресмятане общата стойност на всички фактури в списъка `invoices`. Методът връща пресметнатата стойност.

   (4 точки)

   c) **Метод**

   `public String showSortedByPriceInvoices()`

   Използва задължително Stream API за сортиране на `invoices` в низходящ ред на общата цена на всяка фактура. Методът връща резултата от сортирането като `TextBlock` от вида (вижте примерното решение)

```
The invoices sorted by total price of invoice:
Price:
Invoice:
Price:
Invoice:  .
…….
```

   (5 точки)

   d) **Метод**

   `public String printTotalByBarcode ()`

   Обхожда всички `invoices` и намира колко пъти се среща всеки `BARCODE` в списъка `items` на всяка фактура. Връща `String`, където всеки `BARCODE` е изписан на отделен ред заедно броя пъти, с които се среща в `invoices`. Редовете с `BARCODE` в изходния `String` да са в низходящ ред на броя пъти (вижте примерното решение)

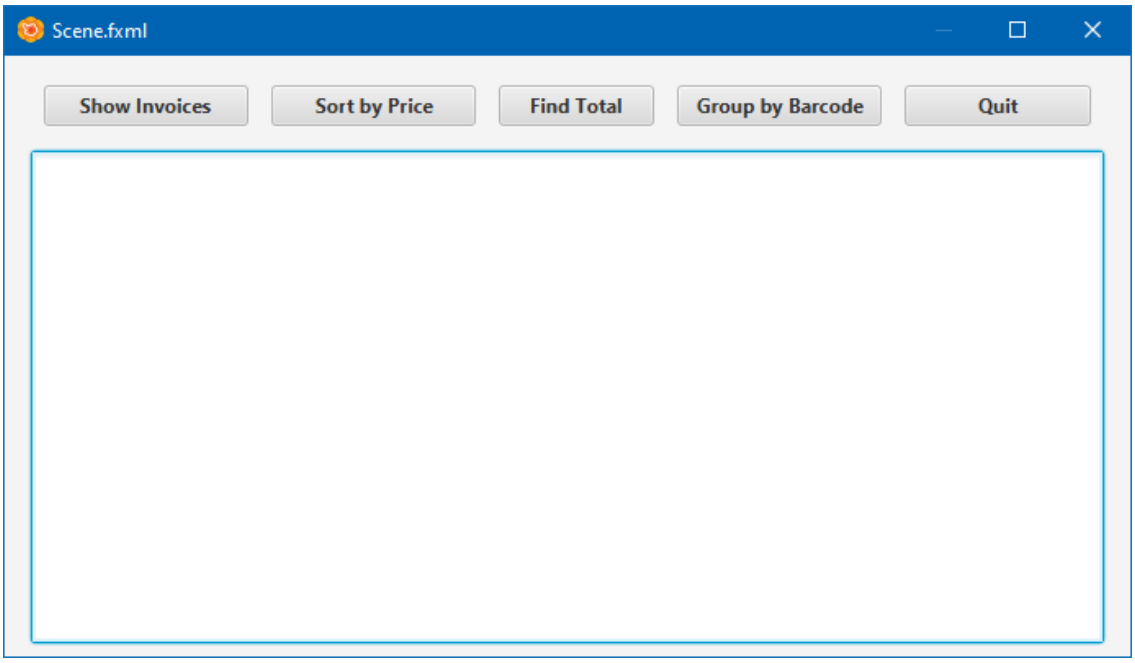
   (8 точки)

Общо точки: 21

В. Добавете нов Java модул именуван като **`exam.gui`** към същия проект, съответен на модула **`package exam.gui`** и файл **`module-info.java`** с описание на JavaFX модул. **Създайте в пакета `exam.gui` следните** FXML артефакти **(FXML сцена, съответни класна контролер и приложение) -**

Точки: 8

1. **Създайте FXML сцена** която да възпроизвежда следния графичен модел, като използвате смислени имена за идентификатори по стила на т. нар. Модифицирана Унгарската нотация (за улеснение в края на текста е дадена примерна структура на JavaFX възлите)



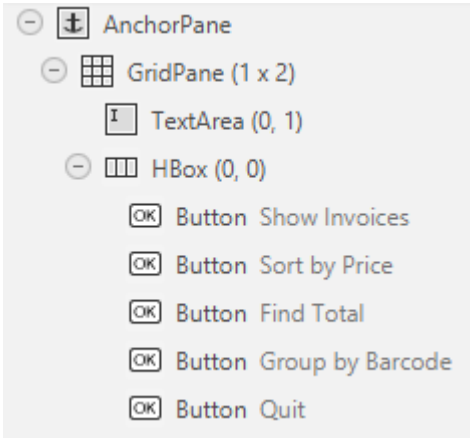
Точки: 10

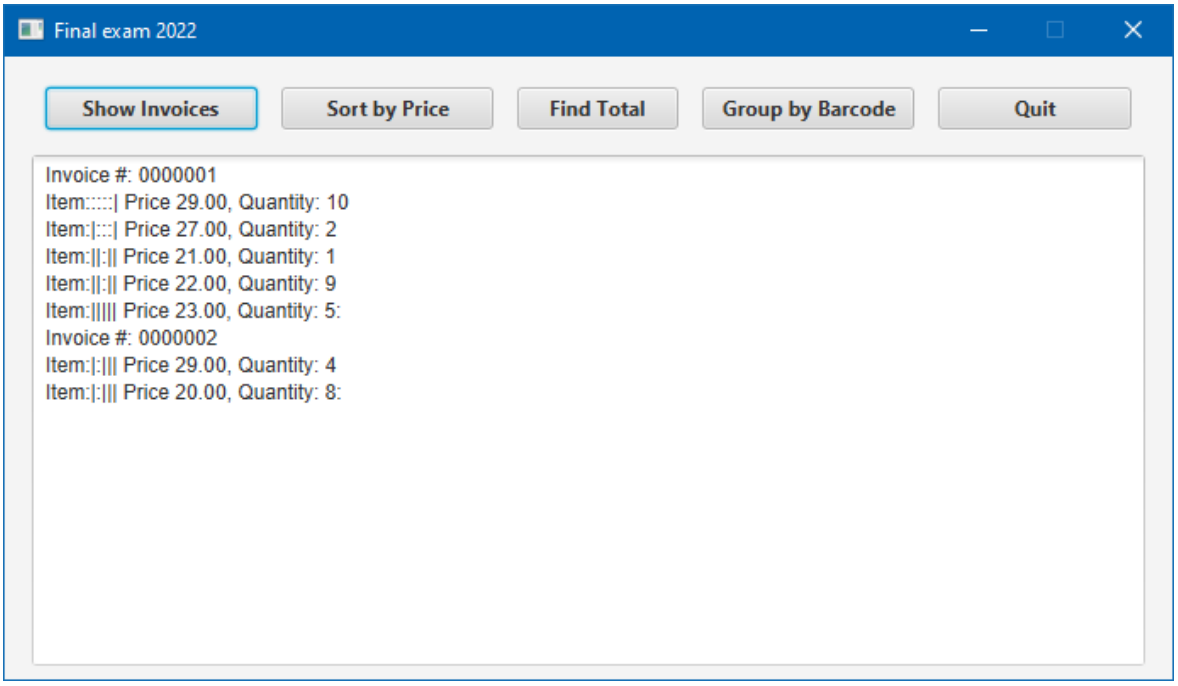
6. **Генерирайте съдържание на Контролера, съответно на тази Сцена. Добавете данна** `invoiceProcessing` **от тип** `InvoiceProcessing` **в Контролера на FXML приложението. Актуализирайте описанието на модулите** `exam.logic` **и** `exam.gui`**, което позволява импортирането на клас** `InvoiceProcessing` **от модул** `exam.logic` **Инициализирайте** `invoiceProcessing` **в метода** `initialize()` **на Контролера, което позволява създаване на** 2 фактури**.**

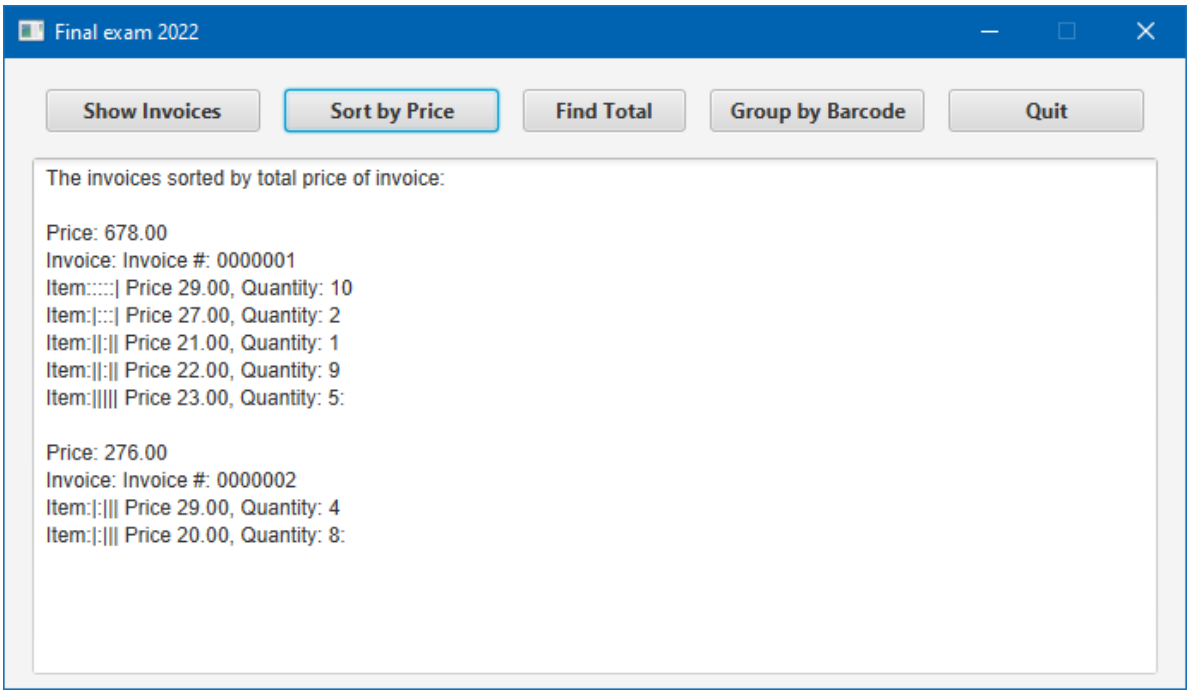
Точки: 5

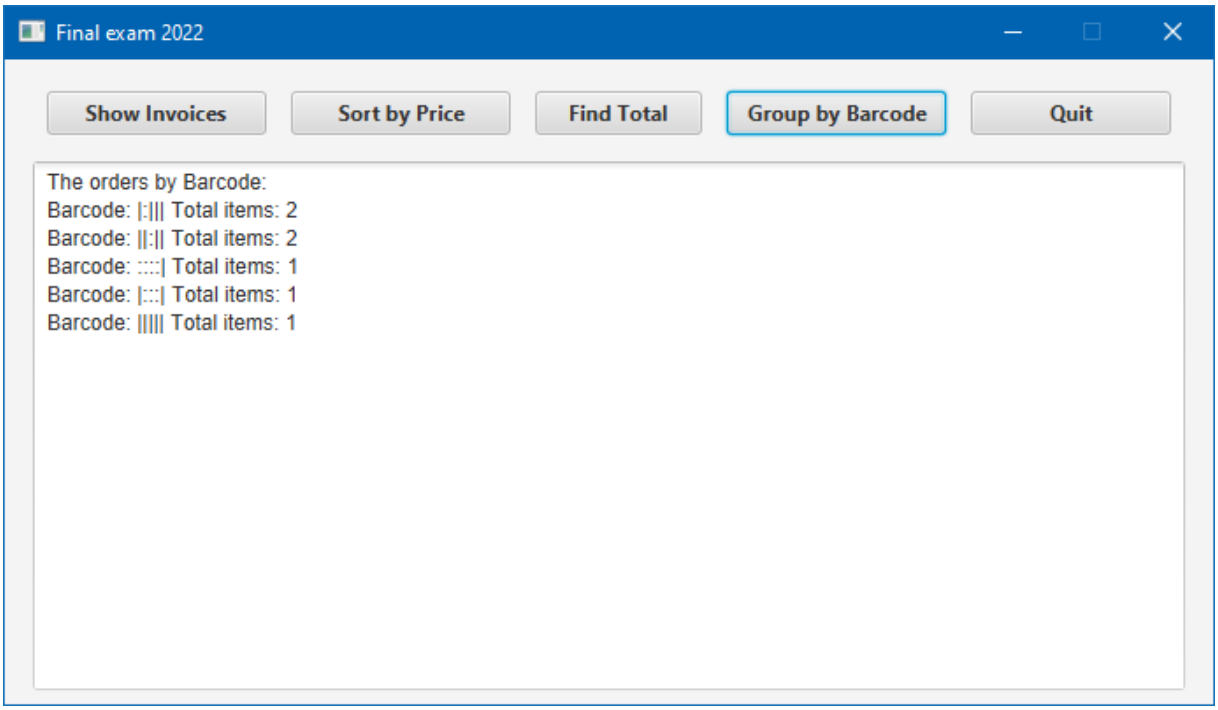
7. **Напишете следната обработка на събитията, създавани при натискане на бутоните на графичния интерфейс (***да се използват методите на* `class InvoiceProcessing`**):**
   - **при натискане на бутона** `Show Invoices` **да се изведе в текстовата област резултата от изпълнението на метода** `showInvoices ()`
   - **при натискане на бутона** `Sort by Price` **да се изведе в текстовата област резултата от изпълнението на метода** `showSortedByPriceInvoices ()`
   - **при натискане на бутона** `Find Total` **да се изведе в текстовата област резултата от изпълнението на метода** `totalPrice()`
   - **при натискане на бутона** `Group by Barcode` **да се изведе в текстовата област резултата от изпълнението на метода** `printTotalByBarcode()`
   - **при натискане на бутона** `Quit` **да се прекрати изпълнението на програмата**

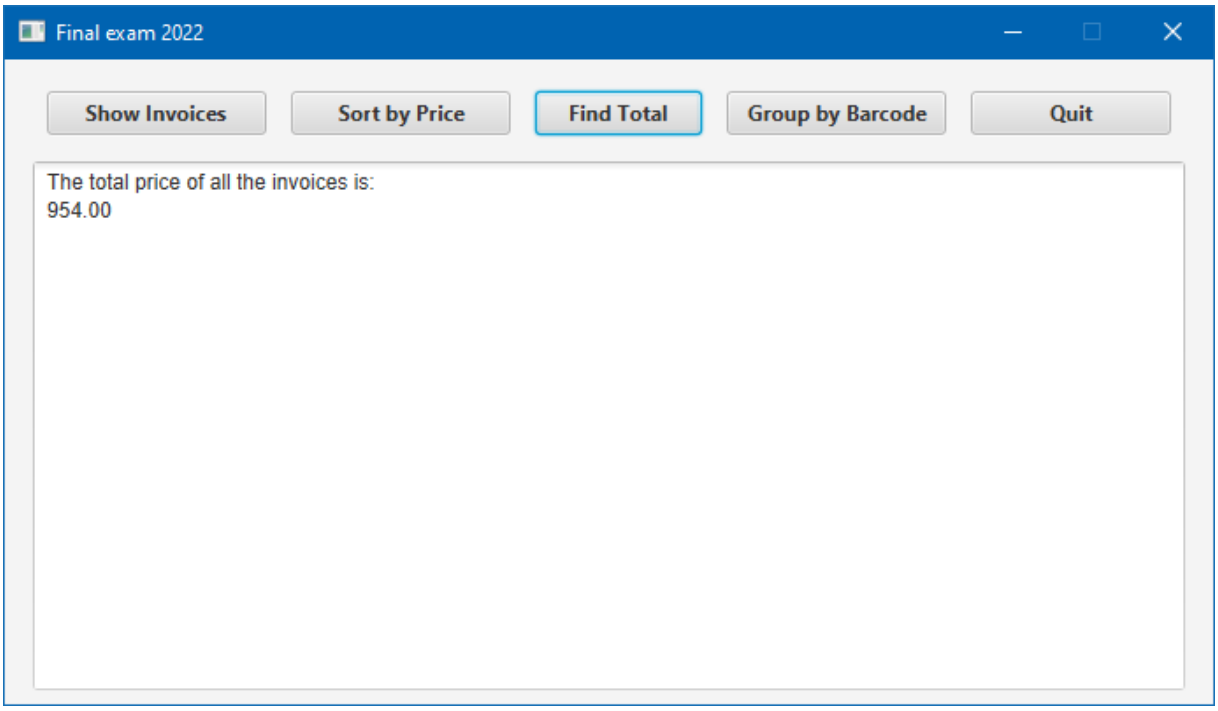
Точки: 10


AnchorPane
GridPane (1 x 2)
TextArea (0, 1)
HBox (0, 0)
Button Show Invoices
Button Sort by Price
Button Find Total
Button Group by Barcode
Button Quit


Final exam 2022
Show Invoices
Sort by Price
Find Total
Group by Barcode
Quit
Invoice #: 0000001
Item:::::| Price 29.00, Quantity: 10
Item:|:::| Price 27.00, Quantity: 2
Item:||:|| Price 21.00, Quantity: 1
Item:||:|| Price 22.00, Quantity: 9
Item:||||| Price 23.00, Quantity: 5:
Invoice #: 0000002
Item:|:||| Price 29.00, Quantity: 4
Item:|:||| Price 20.00, Quantity: 8:


Final exam 2022
Show Invoices
Sort by Price
Find Total
Group by Barcode
Quit
The invoices sorted by total price of invoice:
Price: 678.00
Invoice: Invoice #: 0000001
Item:::::| Price 29.00, Quantity: 10
Item:|:::| Price 27.00, Quantity: 2
Item:||:|| Price 21.00, Quantity: 1
Item:||:|| Price 22.00, Quantity: 9
Item:||||| Price 23.00, Quantity: 5:
Price: 276.00
Invoice: Invoice #: 0000002
Item:|:||| Price 29.00, Quantity: 4
Item:|:||| Price 20.00, Quantity: 8:


Final exam 2022
Show Invoices
Sort by Price
Find Total
Group by Barcode
Quit
The orders by Barcode:
Barcode: |:||| Total items: 2
Barcode: ||:|| Total items: 2
Barcode: ::::| Total items: 1
Barcode: |:::| Total items: 1
Barcode: ||||| Total items: 1


Final exam 2022
Show Invoices
Sort by Price
Find Total
Group by Barcode
Quit
The total price of all the invoices is:
954.00