

|  |
| --- |
| Junior Java Developer  Assessment |
|  |
| March 18  TRG Research and Development  Authored by: Andreas Georgiou GitHub: https://github.com/AGEORGIOU4/Invoicing.git |

# INVOICING JAVA APP

## Documentation

Main

Based on the requirements the application was developed in a procedural programming manner respecting the OOP fundamentals. There are 5 Classes plus the Tests Class for the unit tests. The Account, Customer, Service, Invoice, Item class.

**Structure**

The ‘main’ class is the Customer (first name, middle name, last name, address, phone). Without passing an invoice or a service list from the very beginning, the program allows the creation of an Account (number, Customer) just by adding an account number and injecting a Customer on construction. To achieve this, the Account class has two constructors. The first constructor requires only an account number and a Customer object and the second (number, Customer, list of services, list of invoices). Therefore, the customer is entitled to an empty account. After creating a customer and an account for the program, the application allows the insertion of invoices and items into the account.

An item holds a service and the quantity that the service was used hence a list of items is a collection of services and quantities. An invoice is a collection of items. A user can create many services using the Service class (name, description, price) and populate a vector of services that can be injected or initialized into an Account. Also, each service can be added to a vector of items followed by the quantity ‘sold’. The Customer will be able to get access to the services used through the account. The vector of items (Service, quantity) can calculate the Invoice’s total amount. An invoice has a number, a date created and a list of Items. The account can hold a list of invoices.

**Functionalities**

An account is formed by a number, a customer, a list of services and a list of invoices. The user can generate an Invoice for the customer that will display on the console the Customer’s data, the services used analytically (date of the invoice issuance, services, description, quantity, price), below the auto-calculated balance and lastly each invoice number. The user can add an invoice to the current account or update the previous one. The app will automatically make suitable calculations for both cases.

How to use the app

Main:

1. Create a customer

Customer testCustomer = new Customer("Andreas", "Georgiou", "Georgiou", "Archiepiskopou Makariou C' 111",  
 "+35799131666");

1. Create an account

Account testAccount = new Account(177, testCustomer);

1. Create services

Service testService1 = new Service("Test Service 1", "Simple Service 1", 12);  
Service testService2 = new Service("Test Service 2", "Simple Service 2", 50);

1. Create a vector of services and add the services

Vector<Service> testListOfServices = new Vector<>();  
testListOfServices.add(testService1);  
testListOfServices.add(testService2);

1. Create items

Item testItem1 = new Item(testService1, 2);  
Item testItem2 = new Item(testService2, 3);

1. Create a vector of items and add the items

Vector<Item> testListOfItems = new Vector<>();  
testListOfItems.add(testItem1);  
testListOfItems.add(testItem2);

1. Create an Invoice and inject the list of items

Invoice testInvoice = new Invoice(999, System.*currentTimeMillis*(), testListOfItems);

1. Create a vector of Invoices and add the previous invoice

Vector<Invoice> testListOfInvoices = new Vector<>();  
testListOfInvoices.add(testInvoice);

1. Set services and invoices on the account

testAccount.setServices(testListOfServices);  
testAccount.setInvoices(testListOfInvoices);

1. Generate the invoice

testAccount.generateCustomerInvoice();

==========================================

Name: Andreas Georgiou Georgiou

Address: Archiepiskopou Makariou C' 111

Phone: +35799131666

Services:

Mar 18,2021 04:53 | Test Service 1 | Simple Service 1 | x 2 | 12.0

Mar 18,2021 04:53 | Test Service 2 | Simple Service 2 | x 3 | 50.0

Balance: £174.0

Invoice: #999

1. Create a new invoice with a new service and update the account

Service testService3 = new Service("Test Service 3", "Simple Service 3", 64.50f);  
testListOfServices.add(testService3);  
  
Item testItem3 = new Item(testService3, 3);  
testListOfItems.add(testItem3);  
  
Invoice testInvoice2 = new Invoice(999, System.*currentTimeMillis*(), testListOfItems);  
testListOfInvoices.add(testInvoice2);  
testAccount.setInvoices(testListOfInvoices);

1. Generate updated invoice

testAccount.generateCustomerInvoice();

==========================================

Name: Andreas Georgiou Georgiou

Address: Archiepiskopou Makariou C' 111

Phone: +35799131666

Services:

Mar 18,2021 04:59 | Test Service 1 | Simple Service 1 | x 2 | 12.0

Mar 18,2021 04:59 | Test Service 2 | Simple Service 2 | x 3 | 50.0

Mar 18,2021 04:59 | Test Service 3 | Simple Service 3 | x 3 | 64.5

Mar 18,2021 04:59 | Test Service 1 | Simple Service 1 | x 2 | 12.0

Mar 18,2021 04:59 | Test Service 2 | Simple Service 2 | x 3 | 50.0

Mar 18,2021 04:59 | Test Service 3 | Simple Service 3 | x 3 | 64.5

Balance: £541.5

Invoice: #998, #999

*\*Note: Tests can be performed by running the Tests class*