

Author: **Joan A. Piñol Galofre (japinol)**

Program: **Very Basic Invoice Model Example.**

Version: **0.0.1**

Author gitHub: **japinol7**

Author twitter: **@japinol**

Requirements:

- Code a model for an invoice that:
 - Store the following information:
 - The quantity sold for each product
 - The total amount of the invoice
 - The client information
 - The company information.
 - Simulate storing this information into a database.
- Write unit tests using unittest to test the implementation.
- The implementation will use Python 3.8.

To test the implementation:

You can just go to the project folder, i.e., `erp_invoice_model` and execute the command:

```
$ python -m unittest
```

That should end this way:

```
...
Ran 18 tests in 0.005s
OK
```

Analysis:

I have decided to structure the project in five main addons (module-packages):

`base`, `account`, `contacts`, `product` and `sale`.

Each addon (module-package) will have two packages: `models` and `tests`.

The `account` addon will implement the classes `AccountInvoice` and `AccountInvoiceLine`.

The `base` addon will create a dummy class `DBDriver` to connect and write to a fictional database.

The `sale` addon will implement the classes `SaleInvoice` and `SaleInvoiceLine`. They will be subclasses of `AccountInvoice` and `AccountInvoiceLine` from the `account` addon.

The main tests are in the main addon `account`, inside `test_account_invoice.py` and the `sale` addon, in `test_sale_invoice.py`

In more detail:

- account
 - models
 - *account_invoice.py*
Implements the model *account_invoice*.
Classes: `AccountInvoice`, `AccountInvoiceType` and `AccountInvoiceException`.
It's not intended to be used directly. Instead, its classes are intended to be used through the subclass *sale* addon (and a future *purchase* addon).
 - *account_invoice_line.py*
Implements the model *account_invoice_line*.
Classes: `AccountInvoiceLine`
It's not intended to be used directly. Instead, its classes are intended to be used through the subclass *sale* addon (and a future *purchase* addon).
 - tests
 - *test_account_invoice.py*
Here are the main tests for all the functionality of the invoices.
- base
 - models
 - *db_driver.py*
Implements a dummy model *db_driver*.
Classes: `DBDriver`.
 - tests
 - *test_db_driver.py*
- contacts
 - models
 - *contact.py*
 - tests
 - *test_contact.py*
- product
 - models
 - *product.py*
 - tests
 - *test_product.py*
- sale
 - models
 - *sale_invoice.py*
Implements the model *sale_invoice*.
Classes: `SaleInvoice`.
This class inherits `AccountInvoice` from the *account* addon.
`SaleInvoice` puts the attribute `invoice_type` to `AccountInvoiceType`.
`OUT` and adds the string ' INV/OUT/ ' to the field *invoice_id*.
 - *sale_invoice_line.py*
Implements the model *sale_invoice_line*.
Classes: `SaleInvoiceLine`.
This class inherits `AccountInvoiceLine` from the *account* addon.
 - tests
 - *test_sale_invoice.py*
This addon adds some tests for the characteristics added to sale invoices.
Although, the main tests for invoices are in the file: *test_account_invoice.py*.