Name : MD Mynuddin\ Roll : ASH1825007M

Design Size For ShopAssistant Project

Project Github link = https://github.com/IIT-NSTU/Shop-Assistant

Definition: Design size, measure size in terms of Calculate the:

- Number of Classes
- Number of Interfaces
- Number of Abstract Classes
- Number of Pakcage
- Number of Public Method
- Number of Private Method
- Number of Object
- Number of Public Constructor
- Number of Private Constructor
- Number of Public Variable
- Number of Private Variable in a software project.

Measuring Technique : Automated Programe

In [1]:

```
import pandas as pd

import os
Files = os.listdir('ShopAssistant')
Files
```

Out[1]:

```
['AddCostPage.java',
 'LoginPage.java',
 'CreateSalesInvoicePage.java',
 'DailyIncomeCostPage.java',
 'CreatePurchaseInvoicePage.java',
 'AdvancedStockCheckPage.java',
 'HomePage.java',
 'DueCheckPage.java',
 'StaffAttendancePage.java',
 'SettingsPage.java',
 'StockCheckPage.java']
```

In [2]:

```
print("                               This is for ShopAssistant Java Files
print("------------------------------------------------------------------------------------------------

def Design_Size(Files):

    df = pd.DataFrame(columns = ['File name', 'Number of Classes', 'Number of Interfaces', 'Number of Abstract
    for document in Files:

        if document.endswith(".java"):

            cls = 0
            interface = 0
            abs_class = 0
            package = 0
            pub_meth = 0
            pri_meth = 0
            obj = 0
            pub_cons = 0
            pri_cons = 0
            pub_var = 0
            pri_var = 0

            with open('ShopAssistant/'+document) as file:

                filename = document.partition('.')[0]
                for lines in file:
                    #Class
                    if 'class' in lines:
                        cls = cls + 1

                    #interface
                    if 'interface' in lines:
                        interface = interface + 1

                    #abstract class
                    if 'abstract' in lines:
                        abs_class = abs_class + 1

                    #Package
                    if lines.strip().startswith('import'):
                        package = package + 1

                    #Public Constructor
                    if 'public' in lines and  (filename+'(') in lines and 'new' not in lines:
                        pub_cons = pub_cons + 1
                    #Private Constructor
                    if 'private' in lines and  (filename+'(') in lines and 'new' not in lines:
                        pri_cons = pri_cons + 1

                    #public method
                    datatype = ['void', 'int', 'float', 'double', 'String', 'bool']

                    if 'public' in lines and any(word in lines for word in datatype) and '(' in lines:
                        pub_meth = pub_meth + 1

                    #private method
                    if 'private' in lines and any(word in lines for word in datatype) and '(' in lines:
                        pri_meth = pri_meth + 1

                    #object
                    if 'new' in lines:
                        obj = obj + 1


                    #pub_var
                    class_type = ['private', 'protected']
                    if not any(cls_type in lines for cls_type in class_type) and any(word in lines for word in
                        pub_var = pub_var + 1

                    #pub_var
                    if 'private' in lines and any(word in lines for word in datatype) and not '(' in lines:
                        pri_var = pri_var + 1

                pub_meth = pub_meth - pub_cons
                pri_meth = pri_meth - pri_cons

                list = [filename, cls, interface, abs_class, package, pub_meth, pri_meth, obj, pub_cons, pri_co
                df.loc[len(df)] = list
    return df

import os
Files = os.listdir('ShopAssistant')
Summary = Design_Size(Files)

display(Summary)
```

```
                               This is for ShopAssistant Java Files
-----------------------------------------------------------------------------------------
```

	File name	Number of Classes	Number of Interfaces	Number of Abstract Classes	Number of Pakcage	Number of Public Method	Number of Private Method	Number of Object	Number of Public Constructor	Number of Private Constructor	Number of Public Variable
0	AddCostPage	1	0	0	18	9	0	28	1	0	1
1	LoginPage	1	0	0	13	2	0	19	1	0	0
2	CreateSalesInvoicePage	1	0	0	6	6	0	3	1	0	0
3	DailyIncomeCostPage	1	0	0	18	8	0	44	1	0	14
4	CreatePurchaseInvoicePage	1	0	0	6	6	0	3	1	0	0
5	AdvancedStockCheckPage	1	0	0	19	9	0	43	1	0	10
6	HomePage	1	0	0	18	8	0	30	1	0	9
7	DueCheckPage	1	0	0	23	19	0	57	1	0	10
8	StaffAttendancePage	1	0	0	23	11	0	41	1	0	9
9	SettingsPage	1	0	0	21	19	0	101	1	0	7
10	StockCheckPage	1	0	0	19	5	0	22	1	0	2

In [3]:

```
print("                               This is for Templates Java Files                              ")
print("------------------------------------------------------------------------------------------------
Files = os.listdir('Templates')
def Design_Size(Files):

    df = pd.DataFrame(columns = ['File name', 'Number of Classes', 'Number of Interfaces', 'Number of Abstract
    for document in Files:

        if document.endswith(".java"):

            cls = 0
            interface = 0
            abs_class = 0
            package = 0
            pub_meth = 0
            pri_meth = 0
            obj = 0
            pub_cons = 0
            pri_cons = 0
            pub_var = 0
            pri_var = 0

            with open('Templates/'+document) as file:

                filename = document.partition('.')[0]
                for lines in file:
                    #Class
                    if 'class' in lines:
                        cls = cls + 1

                    #interface
                    if 'interface' in lines:
                        interface = interface + 1

                    #abstract class
                    if 'abstract' in lines:
                        abs_class = abs_class + 1

                    #Package
                    if lines.strip().startswith('import'):
                        package = package + 1

                    #Public Constructor
                    if 'public' in lines and  (filename+'(') in lines and 'new' not in lines:
                        pub_cons = pub_cons + 1
                    #Private Constructor
                    if 'private' in lines and  (filename+'(') in lines and 'new' not in lines:
                        pri_cons = pri_cons + 1

                    #public method
                    datatype = ['void', 'int', 'float', 'double', 'String', 'bool']

                    if 'public' in lines and any(word in lines for word in datatype) and '(' in lines:
                        pub_meth = pub_meth + 1

                    #private method
                    if 'private' in lines and any(word in lines for word in datatype) and '(' in lines:
                        pri_meth = pri_meth + 1

                    #object
                    if 'new' in lines:
                        obj = obj + 1


                    #pub_var
                    class_type = ['private', 'protected']
                    if not any(cls_type in lines for cls_type in class_type) and any(word in lines for word in
                        pub_var = pub_var + 1

                    #pub_var
                    if 'private' in lines and any(word in lines for word in datatype) and not '(' in lines:
                        pri_var = pri_var + 1

                pub_meth = pub_meth - pub_cons
                pri_meth = pri_meth - pri_cons

                list = [filename, cls, interface, abs_class, package, pub_meth, pri_meth, obj, pub_cons, pri_co
                df.loc[len(df)] = list
    return df

import os
Files = os.listdir('Templates')
Summary = Design_Size(Files)

display(Summary)
```

```
                               This is for Templates Java Files
-----------------------------------------------------------------------------------------
```

	File name	Number of Classes	Number of Interfaces	Number of Abstract Classes	Number of Pakcage	Number of Public Method	Number of Private Method	Number of Object	Number of Public Constructor	Number of Private Constructor	Number of Public Variable	Number of Private Variable
0	StartingTemplate	2	0	0	5	2	0	10	1	0	0	0
1	InvoiceGeneratorTemplate	2	0	0	29	44	0	91	1	0	18	0
2	FrameSetup	1	0	0	10	11	0	26	1	0	16	0
3	DashBoardTemplate	2	0	0	22	17	0	35	1	0	2	0

In [4]:

```
print("                               This is for Main Java Files                                   ")
print("------------------------------------------------------------------------------------------------
Files = os.listdir('Main')
def Design_Size(Files):

    df = pd.DataFrame(columns = ['File name', 'Number of Classes', 'Number of Interfaces', 'Number of Abstract
    for document in Files:

        if document.endswith(".java"):

            cls = 0
            interface = 0
            abs_class = 0
            package = 0
            pub_meth = 0
            pri_meth = 0
            obj = 0
            pub_cons = 0
            pri_cons = 0
            pub_var = 0
            pri_var = 0

            with open('Main/'+document) as file:

                filename = document.partition('.')[0]
                for lines in file:
                    #Class
                    if 'class' in lines:
                        cls = cls + 1

                    #interface
                    if 'interface' in lines:
                        interface = interface + 1

                    #abstract class
                    if 'abstract' in lines:
                        abs_class = abs_class + 1

                    #Package
                    if lines.strip().startswith('import'):
                        package = package + 1

                    #Public Constructor
                    if 'public' in lines and  (filename+'(') in lines and 'new' not in lines:
                        pub_cons = pub_cons + 1
                    #Private Constructor
                    if 'private' in lines and  (filename+'(') in lines and 'new' not in lines:
                        pri_cons = pri_cons + 1

                    #public method
                    datatype = ['void', 'int', 'float', 'double', 'String', 'bool']

                    if 'public' in lines and any(word in lines for word in datatype) and '(' in lines:
                        pub_meth = pub_meth + 1

                    #private method
                    if 'private' in lines and any(word in lines for word in datatype) and '(' in lines:
                        pri_meth = pri_meth + 1

                    #object
                    if 'new' in lines:
                        obj = obj + 1


                    #pub_var
                    class_type = ['private', 'protected']
                    if not any(cls_type in lines for cls_type in class_type) and any(word in lines for word in
                        pub_var = pub_var + 1

                    #pub_var
                    if 'private' in lines and any(word in lines for word in datatype) and not '(' in lines:
                        pri_var = pri_var + 1

                pub_meth = pub_meth - pub_cons
                pri_meth = pri_meth - pri_cons

                list = [filename, cls, interface, abs_class, package, pub_meth, pri_meth, obj, pub_cons, pri_co
                df.loc[len(df)] = list
    return df

import os
Files = os.listdir('Main')
Summary = Design_Size(Files)

display(Summary)
```

```
                               This is for Main Java Files
-----------------------------------------------------------------------------------------
```

	File name	Number of Classes	Number of Interfaces	Number of Abstract Classes	Number of Pakcage	Number of Public Method	Number of Private Method	Number of Object	Number of Public Constructor	Number of Private Constructor	Number of Public Variable	Number of Private Variable
0	MainPage	1	0	0	9	2	0	19	0	0	0	0

In [ ]:

In [ ]: