In [1]:

In [2]:

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 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',

This is for ShopAssistant Java Files

'DueCheckPage.java', 'StaffAttendancePage.java',

df = pd.DataFrame(columns = ['File name', 'Number of Classes', 'Number of Interfaces', 'Number of Abstract

'SettingsPage.java', 'StockCheckPage.java'] print(" def Design Size(Files): for document in Files: if document.endswith(".java"):

cls = 0

interface = 0 abs class = 0package = 0 pub meth = 0pri meth = 0

obj = 0 pub cons = 0pri_cons = 0 pub var = 0

pri var = 0

for lines in file: #Class

if 'class' in lines: cls = cls + 1

with open('ShopAssistant/'+document) as file: filename = document.partition('.')[0] #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

#public method

#private method

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

#object

#pub var

#pub var

pri_cons = pri_cons + 1

pub_meth = pub_meth + 1

pri_meth = pri_meth + 1

class_type = ['private', 'protected']

This is for ShopAssistant Java Files

Number

Pakcage

of

18

13

18

6

18

23

23

21

This is for Templates Java Files

df = pd.DataFrame(columns = ['File name', 'Number of Classes', 'Number of Interfaces', 'Number of Abstract

if 'public' in lines and (filename+'(') in lines and 'new' not in lines:

if 'private' in lines and (filename+'(') in lines and 'new' not in lines:

if 'public' in lines and any(word in lines for word in datatype) and '(' in lines:

if 'private' in lines and any (word in lines for word in datatype) and '(' in lines:

if not any (cls_type in lines for cls_type in class_type) and any (word in lines for word in

if 'private' in lines and any (word in lines for word in datatype) and not '(' in lines:

list = [filename, cls, interface, abs_class, package, pub_meth, pri_meth, obj, pub_cons, pri_cd

Number Number

Public

Method

2

11

df = pd.DataFrame(columns = ['File name', 'Number of Classes', 'Number of Interfaces', 'Number of Abstract

if 'public' in lines and (filename+'(') in lines and 'new' not in lines:

if 'private' in lines and (filename+'(') in lines and 'new' not in lines:

if 'public' in lines and any (word in lines for word in datatype) and '(' in lines:

if 'private' in lines and any (word in lines for word in datatype) and '(' in lines:

if not any(cls_type in lines for cls_type in class_type) and any(word in lines for word in

if 'private' in lines and any(word in lines for word in datatype) and not '(' in lines:

list = [filename, cls, interface, abs_class, package, pub_meth, pri_meth, obj, pub_cons, pri_colors, p

Number of

of

19

Public

Object Constructor Constructor

0

Number of

Private

Number Number of

Private

Variable

0

of Public

Variable

Number Number of Number

Private

Method

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

Number

of

10

26

of

0

Private

Method

Number of

Public

Object Constructor Constructor

Number of

Private

0

0

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

Number

Abstract

Classes

of

0

0

0

0

Number

of

0

0

0

0

0

0

pub_var = pub_var + 1

pri_var = pri_var + 1

pub_meth = pub_meth - pub_cons pri_meth = pri_meth - pri_cons

df.loc[len(df)] = list

Number

of

1

1

1

1

1

1

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

if 'class' in lines: cls = cls + 1

if 'interface' in lines:

if 'abstract' in lines:

#Public Constructor

#Private Constructor

#public method

#private method

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

#object

#pub var

interface = interface + 1

abs_class = abs_class + 1

package = package + 1

pub cons = pub cons + 1

pri_cons = pri_cons + 1

pub_meth = pub_meth + 1

pri_meth = pri_meth + 1

class_type = ['private', 'protected']

This is for Templates Java Files

Number

Pakcage

of

5

29

10

22

This is for Main Java Files

Number

Abstract

Classes

of

0

0

Number

of

0

0

pub_var = pub_var + 1

pri_var = pri_var + 1

pub meth = pub meth - pub cons pri_meth = pri_meth - pri_cons

df.loc[len(df)] = list

Number

of

2

1

Classes Interfaces

return df

display(Summary)

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

File name

StartingTemplate

DashBoardTemplate

Files = os.listdir('Main') def Design_Size(Files):

for document in Files:

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

if document.endswith(".java"):

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

if 'class' in lines: cls = cls + 1

if 'interface' in lines:

if 'abstract' in lines:

#Public Constructor

#Private Constructor

#public method

#private method

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

#object

#pub var

#pub var

interface = interface + 1

abs_class = abs_class + 1

package = package + 1

pub cons = pub cons + 1

pri_cons = pri_cons + 1

pub_meth = pub_meth + 1

pri_meth = pri_meth + 1

class type = ['private', 'protected']

This is for Main Java Files

of

of Public

Method

2

pub var = pub var + 1

pri_var = pri_var + 1

Number of Number

0

Classes Pakcage

Abstract

pub_meth = pub_meth - pub_cons pri_meth = pri_meth - pri_cons

df.loc[len(df)] = list

return df

display(Summary)

File

name

0 MainPage

In []:

In []:

Files = os.listdir('Main') Summary = Design Size(Files)

Number

of

Classes Interfaces

Number

of

0

import os

if lines.strip().startswith('import'):

for lines in file: #Class

#interface

#Package

#abstract class

filename = document.partition('.')[0]

FrameSetup

InvoiceGeneratorTemplate

import os

0

2

print("

print("----

In [4]:

if lines.strip().startswith('import'):

for lines in file: #Class

#interface

#abstract class

filename = document.partition('.')[0]

Classes Interfaces

return df

display(Summary)

Files = os.listdir('ShopAssistant')

File name

AddCostPage

CreateSalesInvoicePage

4 CreatePurchaseInvoicePage

AdvancedStockCheckPage

DailyIncomeCostPage

LoginPage

HomePage

DueCheckPage

SettingsPage

StockCheckPage

Files = os.listdir('Templates')

for document in Files:

cls = 0

interface = 0 abs class = 0 package = 0 pub meth = 0pri_meth = 0 obj = 0pub cons = 0pri_cons = 0 pub var = 0pri_var = 0

if document.endswith(".java"):

def Design_Size(Files):

StaffAttendancePage

Summary = Design_Size(Files)

import os

0

1

2

3

6

7

8

9

10

print("

In [3]:

if 'private' in lines and (filename+'(') in lines and 'new' not in lines:

if 'public' in lines and any (word in lines for word in datatype) and '(' in lines:

if 'private' in lines and any (word in lines for word in datatype) and '(' in lines:

if not any(cls_type in lines for cls_type in class_type) and any(word in lines for word in

if 'private' in lines and any(word in lines for word in datatype) and not '(' in lines:

list = [filename, cls, interface, abs_class, package, pub_meth, pri_meth, obj, pub_cons, pri_co

Number Number

of

9

6

8

6

19

11

19

5

Public

Method

Number

of

28

19

44

57

41

101

22

of

0

0

0

Private

Method

Number of

Public

Object Constructor Constructor

1

1

1

1

1

1

Number of

Private

0

0

0

0

0

Number Numb

Priva

Variak

of

0

0

14

0

10

9

7

2

Number Number

Private

Variable

0

of

0

18

16

Public

Variable

")

Public

Variable

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