

---

# ASCETIC

Automated Code Smell Identification and Correction

---

## TEST EXECUTION REPORT

VERSION 1.0



5 febbraio 2019

Coordinatore Progetto:

Nome	Matricola
Manuel De Stefano	0522500633

Partecipanti:

Nome	Matricola
Amoriello Nicola	0512104742
Di Dario Dario	0512104758
Gambardella Michele Simone	0512104502
Iovane Francesco	0512104550
Pascucci Domenico	0512102950
Patierno Sara	0512103460

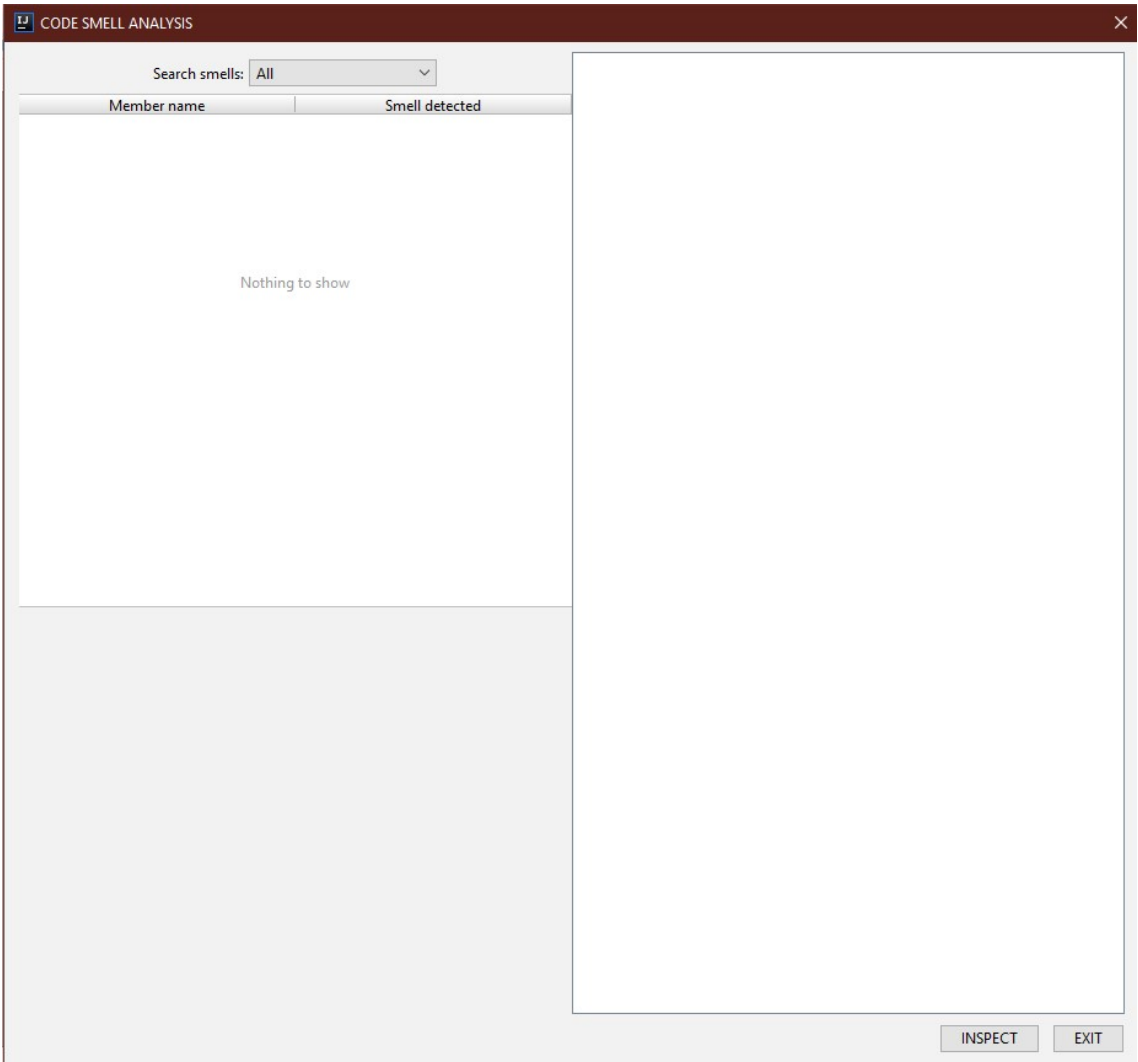
Revision History:

Data	Versione	Descrizione	Autore
04/02/2019	1.0	Prima stesura	Tutto il Team

1    Analisi Code Smells

Nome	TC_0.1_1
Data e ora	04/02/2019 15:45
Tester	Francesco
Output atteso	L'analisi non è andata a buon fine perché non vengono presentati all'utente nessun elenco di codeSmell.
Output del sistema	Nessun output. Il sistema mostra una schermata priva d'informazioni.
Esito	Corretto
Numero Prove	3
Categoria	SA2, PA1

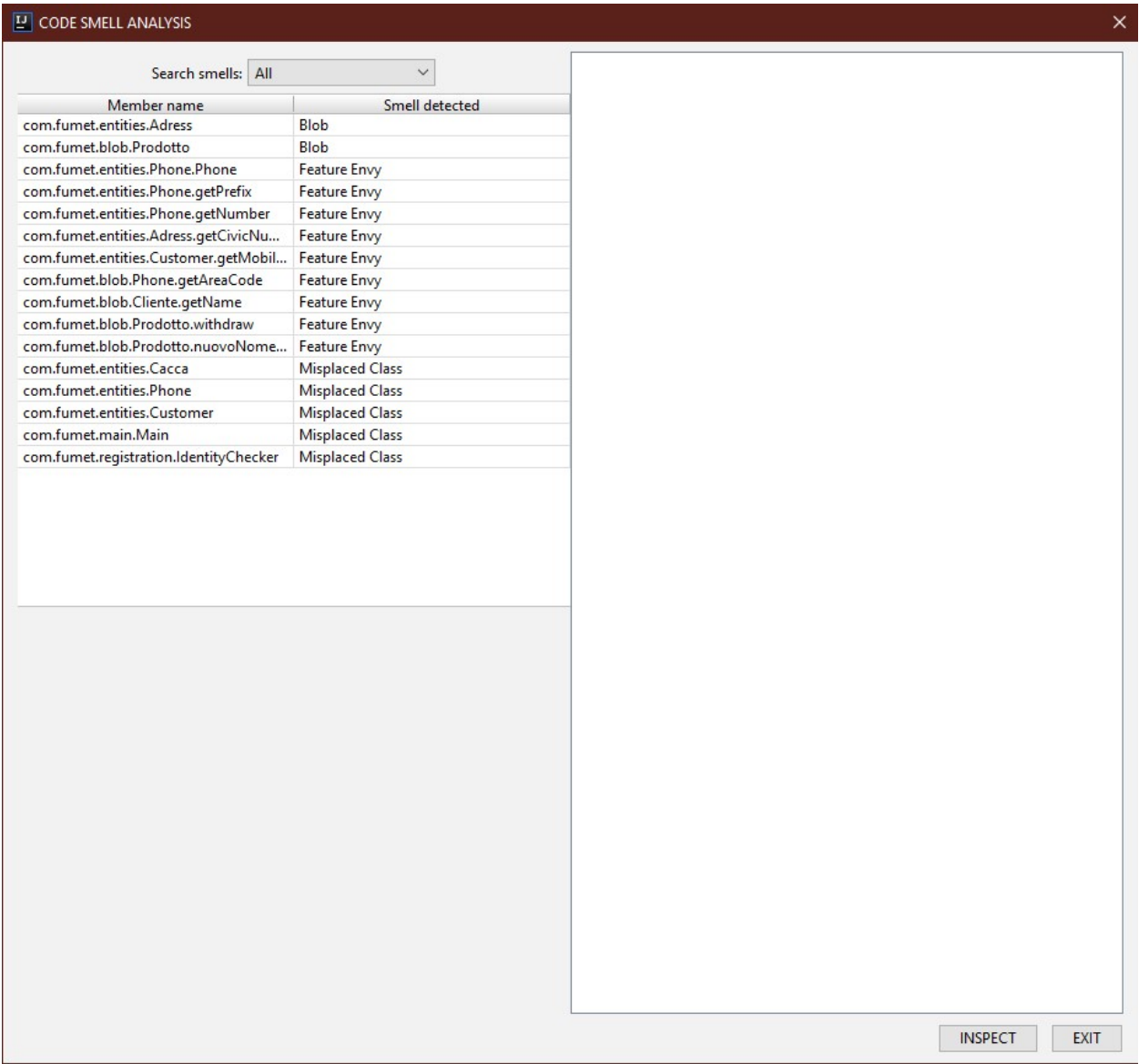
1.1    Dimostrazione Analisi Code Smells



Schermata iniziale di analisi dei codeSmells

Nome	TC_0.1_2
Data e ora	04/02/2019 15:55
Tester	Francesco
Output atteso	L'analisi è andata a buon fine perché viene presentata all'utente una lista di codeSmell.
Output del sistema	In output viene presentato un frame con tutti i codeSmell analizzati dal programma.
Esito	Corretto
Numero Prove	3
Categoria	SA1, PA3

1.2 Dimostrazione Analisi Code Smells

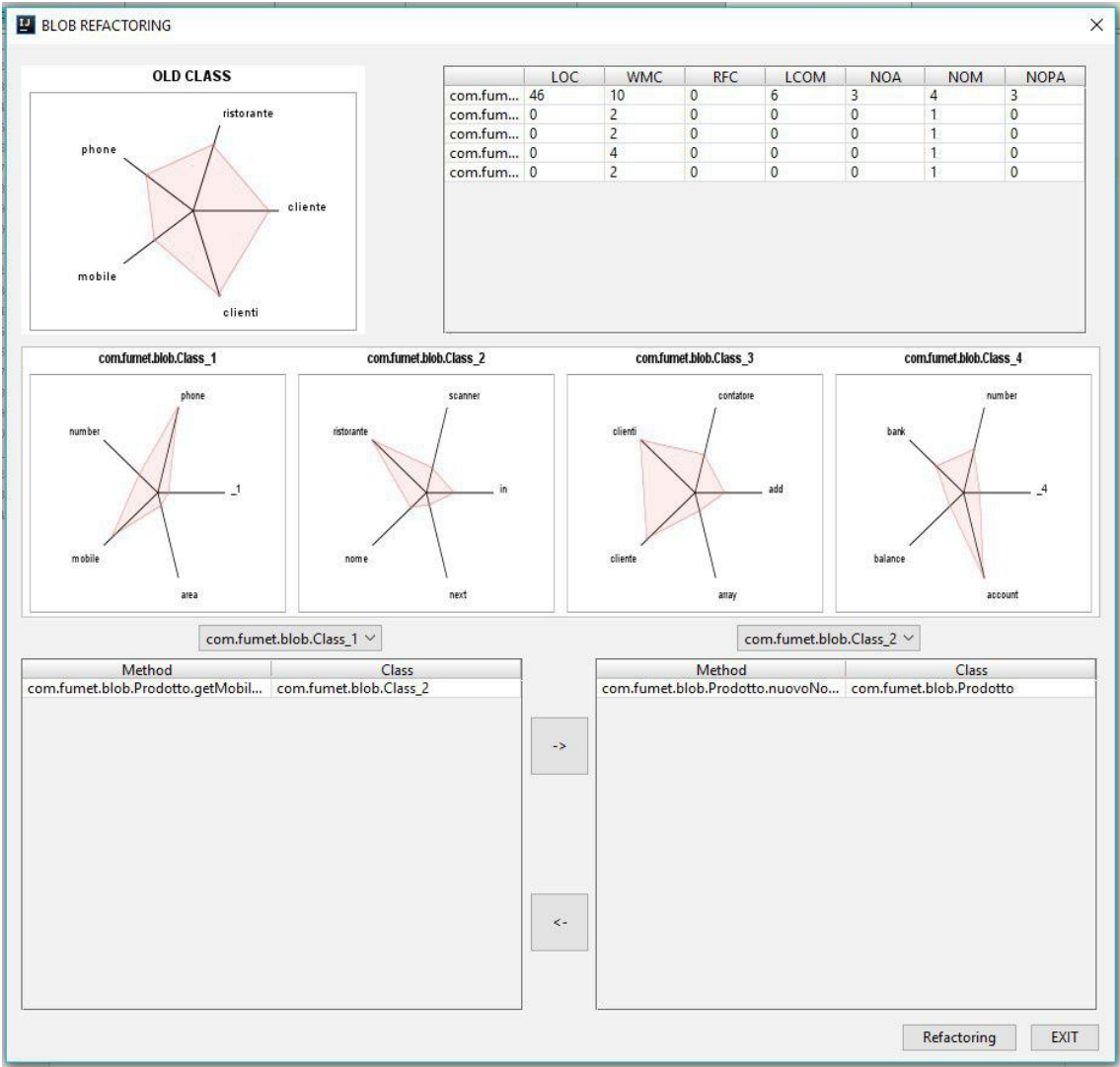


Schermata iniziale di analisi dei codeSmells

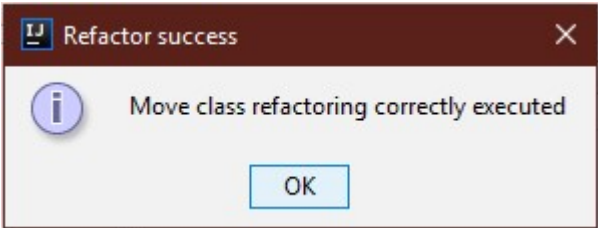
2 Refactor Blob

Nome	TC_1.1_1
Data e ora	04/02/2019 16:07
Tester	Francesco
Output atteso	Il refactor del blob è andato a buon fine perché è stato estratto l’elemento codeSmell dalla classe di origine.
Output del sistema	Il sistema mostra il messaggio: "Blob Corrected, check new classes generated name".
Esito	Corretto
Numero Prove	1
Categoria	AB1, SPF1, SPS1

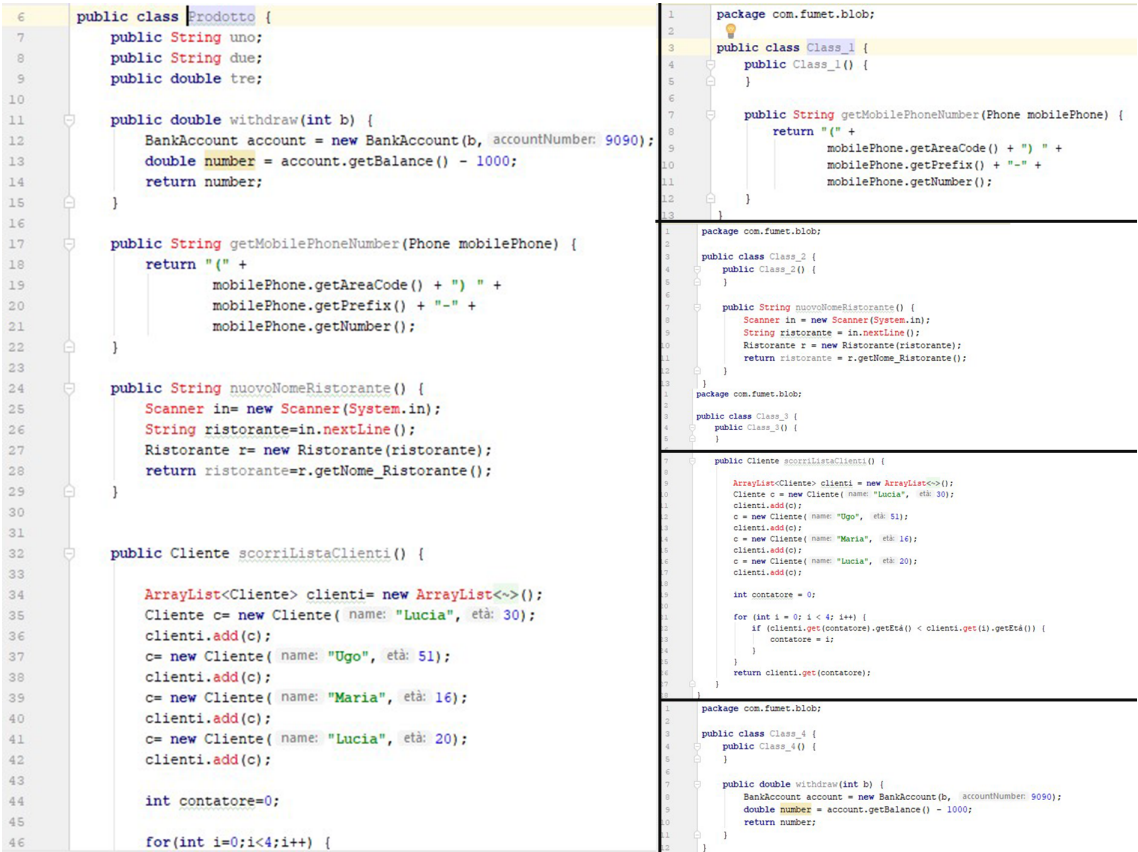
2.1 Dimostrazione Refactor Blob



Schermata iniziale di refactoring per Blob



Feedback di riuscito refactor

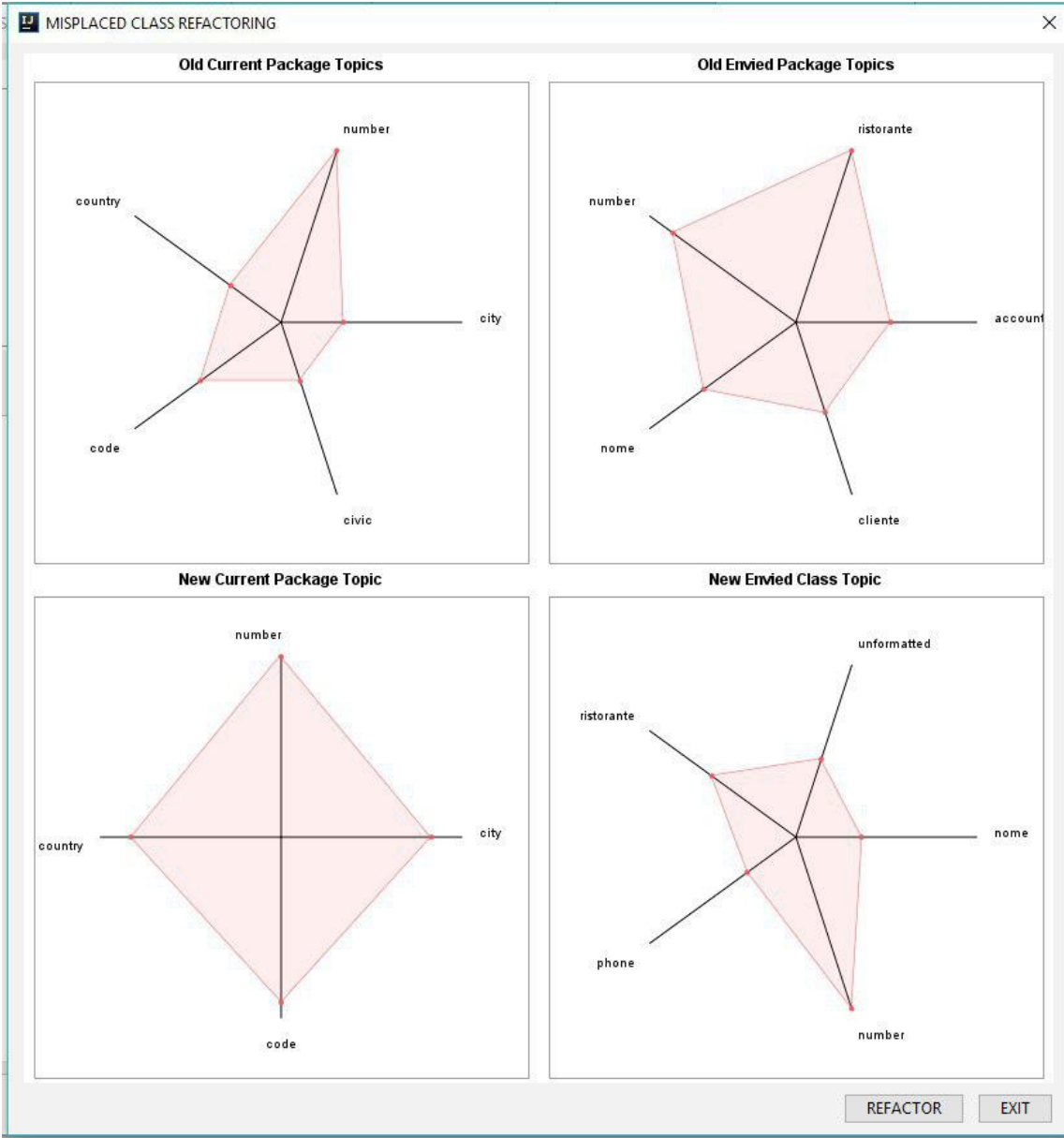


Confronto classe prima/dopo refactoring

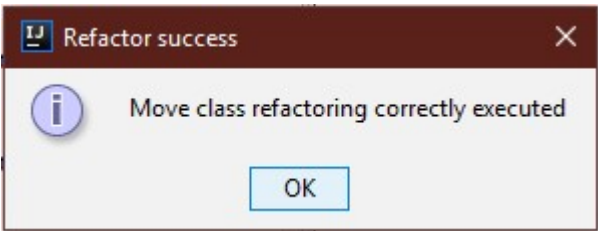
3 Refactor MisplacedClass

Nome	TC_2.1_1
Data e ora	04/02/2019 16:20
Tester	Gelsomino
Output atteso	Il refactor del MisplacedClass è andato a buon fine poichè la classe affetta è stata spostata nel package di destinazione.
Output del sistema	Il sistema mostra il messaggio: "Move class refactoring correctly executed".
Esito	Corretto
Numero Prove	2
Categoria	CA1, PD1

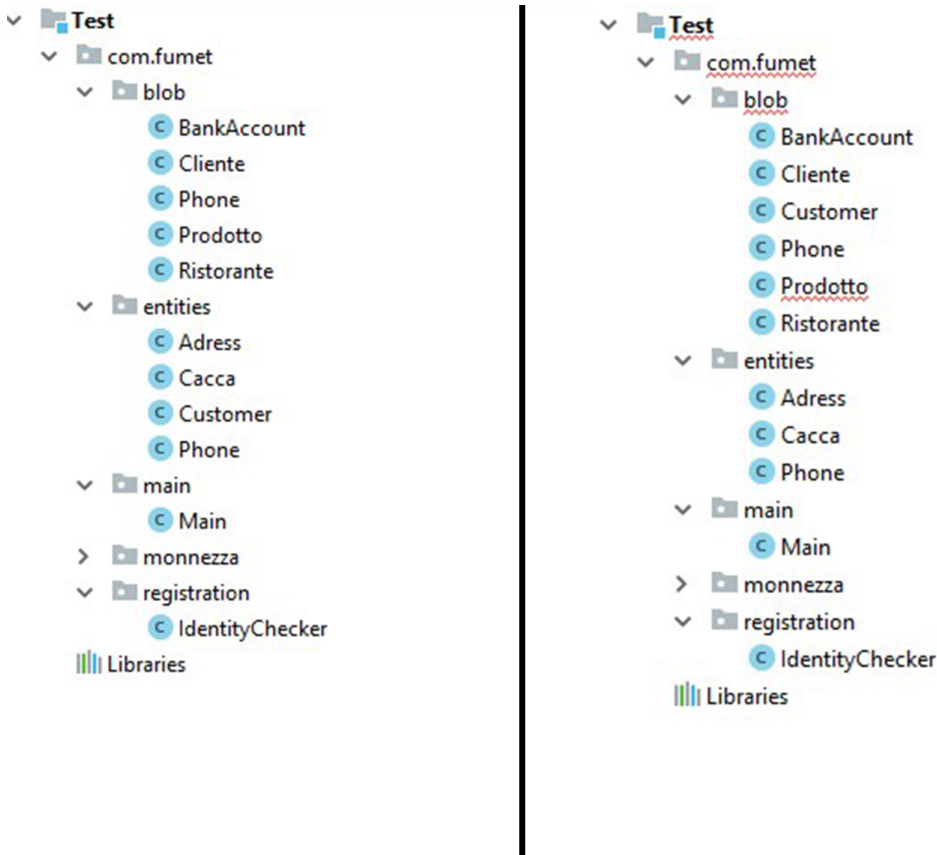
3.1 Dimostrazione Refactor MisplacedClass



Schermata iniziale di refactoring per MisplacedClass



Feedback di riuscito refactor



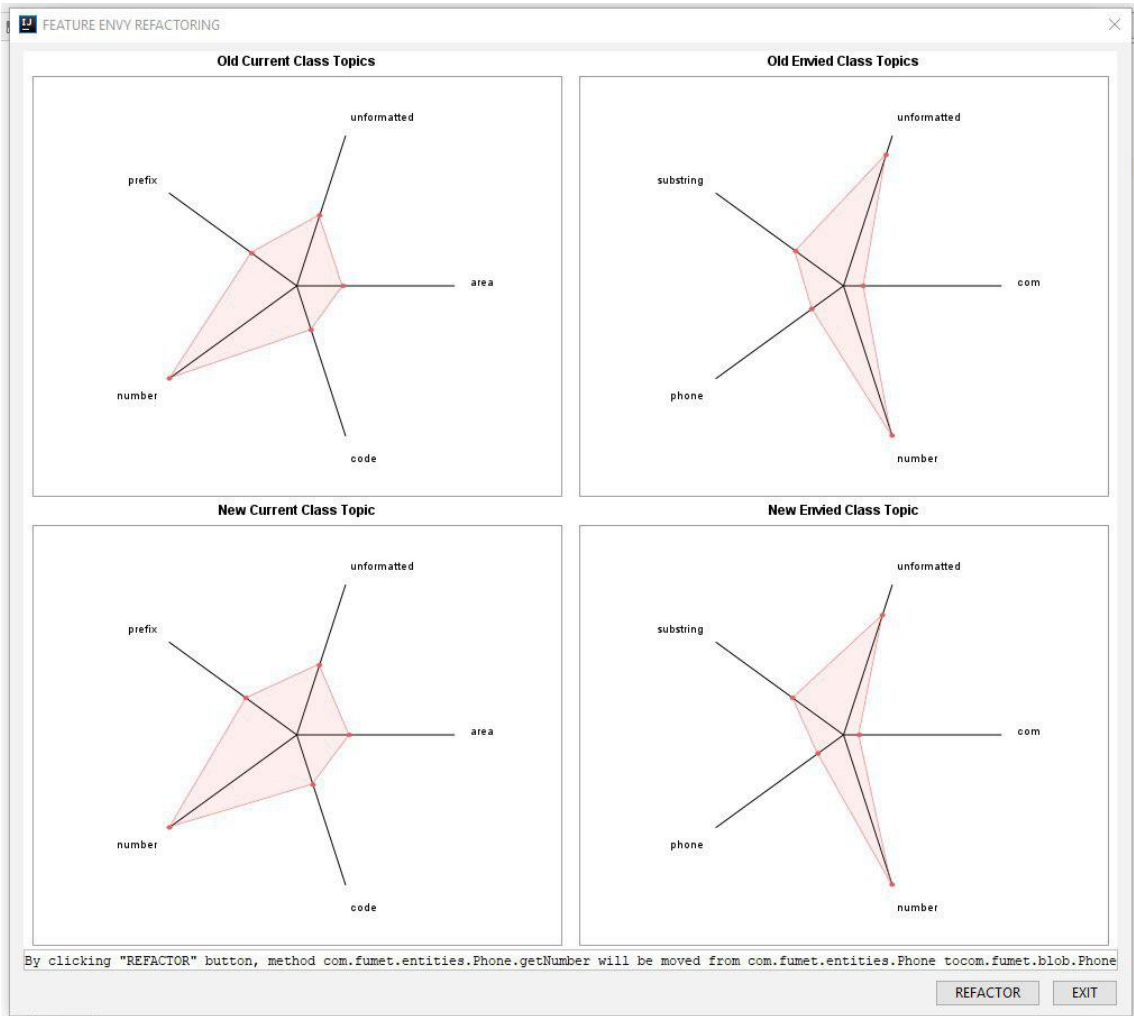
Confronto classe prima/dopo refactoring



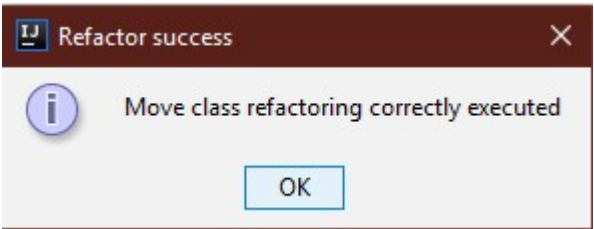
4 Refactor FeatureEnvy

Nome	TC_3.1_1
Data e ora	04/02/2019 16:33
Tester	Francesco
Output atteso	Il refactor del FeatureEnvy è andato a buon fine poichè il metodo affetto è stato spostato nella classe di destinazione.
Output del sistema	Il sistema mostra il messaggio: "Move method refactoring correctly executed".
Esito	Corretto
Numero Prove	3
Categoria	AC1, CD1

4.1 Dimostrazione Refactor FeatureEnvy



Schermata iniziale di refactoring per FeatureEnvy



Feedback di riuscito refactor

```
1 package com.funet.entities;
2
3 public class Customer {
4
5     private String name;
6     private Phone tel;
7
8     public Customer(String name)
9     {
10         this.name=name;
11     }
12
13     public String getName()
14     {
15         return name;
16     }
17
18     public String getMobilePhoneNumber(Phone p) {
19         return tel.getMobilePhoneNumber(p);
20     }
21 }
```

```
1 package com.funet.entities;
2
3 public class Phone {
4
5     private final String unformattedNumber;
6     private final String operator;
7     public Phone(String unformattedNumber) {
8         this.unformattedNumber = unformattedNumber;
9         this.operator = "phone operator";
10    }
11
12    public String getAreaCode() { return unformattedNumber.substring(0,3); }
13
14    public String getPrefix() { return unformattedNumber.substring(3,6); }
15
16    public String getNumber() { return unformattedNumber.substring(6,10); }
17
18    public String getOperator() { return operator;getAreaCode()+getPrefix()+getNumber(); }
19
20    public String reverseNumber() { return getNumber().getPrefix().getAreaCode(); }
21
22    public String italianNumber() { return "39"+getNumber()+getPrefix()+getAreaCode(); }
23
24    public boolean pushNumber() {
25        if(getAreaCode().equals(getPrefix()))
26        {
27            if(getPrefix().equals(getNumber())){return false;}
28            else{ unformattedNumber.replace(getNumber(),"lol");
29                return true;
30            }
31        }
32        else{return false;}
33    }
34
35    public String getMobilePhoneNumber(Phone p) {
36        return getAreaCode() + " " +
37            getPrefix() + "-" +
38            getNumber();
39    }
40 }
```

Confronto classe prima/dopo refactoring