

# INGEGNERIA DEL SOFTWARE: MONOPOLY

Documentazione del progetto “Elaborato 2014/2015”

Università degli Studi di Brescia

Dipartimento di Ingegneria dell'Informazione

Via Branze 38

25231- Brescia



## Relazione dell'elaborato, parte I

---

### Autori

*Nome: Falletti Davide*

*Matricola: 85973*

*Nome: Mele Fabrizio*

*Matricola: 85971*

*Nome: Cordioli Francesco*

*Matricola: 86538*

# Sommario

---

1 Introduzione.....	4
2 Strumenti applicativi utilizzati.....	4
3 Modello di processo.....	4
4 Manuale d'uso.....	5
5 Release 1.....	9
5.1 Diagramma UML delle classi.....	9
5.2 Casi d'uso.....	10
6 Release 2.....	12
6.1 Diagramma UML delle classi.....	12
6.2 Casi d'uso.....	13
7 Release 3.....	15
7.1 Diagramma UML delle classi.....	15
7.2 Casi d'uso.....	16

8 Release 4.....	19
8.1 Diagramma UML delle classi.....	19
8.2 Casi d'uso.....	20
9 Release 5.....	23
9.1 Diagramma UML delle classi.....	23
9.2 Casi d'uso.....	24
10 Testing.....	27

# Ingegneria del Software

---

Monopoly

## Introduzione

---

Il progetto ha l'obiettivo di realizzare tramite un processo di sviluppo incrementale un'applicazione software che consenta di giocare a Monopoly.

## Strumenti applicativi utilizzati

---

Linguaggio di programmazione:	Java
Ambiente di sviluppo:	Eclipse
Strumenti di condivisione:	GitHub, Sourcetree

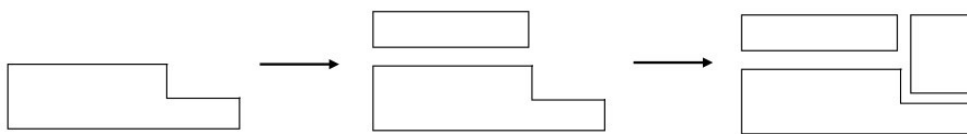
## Modello di processo

---

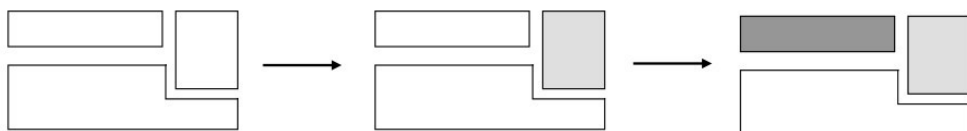
Il modello di processo adottato per lo sviluppo del software è stato il modello Incrementale/Iterativo. Questo modello consente di creare più versioni del software, ognuna delle quali integra un maggior numero di requisiti rispetto alla versione precedente.

Le varie fasi possono essere rappresentate da questo modello:

Approccio incrementale:



Approccio Iterativo:



# Manuale d'uso

## Installazione

L'applicazione è stata scritta nel linguaggio di programmazione OO Java, quindi è eseguibile su qualsiasi calcolatore che abbia installato JRE (*Java Runtime Environment*).

## Esecuzione

Per l'avvio dell'applicazione (.jar) eseguire i seguenti passaggi:

Aprire il Terminale di sistema ( Shell di Unix o Prompt di Windows, a seconda del Sistema Operativo)

- Portarsi nella cartella in cui è presente il file **Progetto\_Exe.jar**
- Eseguire il file .jar mediante il comando `"java -jar Progetto_Exe.jar"`

A questo punto, se non vengono visualizzati errori a causa di una errata versione di JRE, dovrebbe presentarsi questa situazione

## Comandi

```
~~~~~  
~ BENVENUTO NEL GIOCO DEL MONOPOLY ~  
~~~~~
```

```
-----  
Menu' Monopoly  
-----  
1      Inserisci Nuovo Giocatore  
2      Stampa Elenco Giocatori  
3      Gioca  
4      Rimuovi Giocatore  
5      Reset  
  
0      Esci
```

Digita il numero dell'opzione desiderata >

Nome	Inserimento nuovo giocatore
Numero	1
Funzione	L'applicazione permette di inserire il nome di un nuovo giocatore.
Immagini	<pre>Digita il numero dell'opzione desiderata &gt; 1 Inserisci un nuovo giocatore &gt; Dab</pre>
Note	Se è già stato inserito il numero massimo di giocatori, il programma impedisce l'inserimento di un nuovo giocatore e lo comunica all'utente.

Nome	Stampa elenco giocatori
Numero	2
Funzione	L'applicazione permette di visualizzare l'elenco di tutti i giocatori che sono stati inseriti. caricato!il!modello!nel!sistema!
Immagini	! <p> Digita il numero dell'opzione desiderata &gt; 2  1 - Dab[0e]  2 - fab[0e]  3 - cioppo[0e] </p>
Note	Se non ci sono giocatori inseriti, il programma lo comunica all'utente.

Nome	Gioca
Numero	3
Funzione	L'applicazione permette di giocare una partita a Monopoly, permettendo l'avanzamento dei turni tramite la pressione del tasto invio
Immagini	! <p> Digita il numero dell'opzione desiderata &gt; 3  Turno 1  ***** </p> <p> [cioppo] lancia i dadi ed escono: 2 e 6.  Si muove da Via![0] a Viale Monterosa[8].  [cioppo] ha acquistato Viale Monterosa  Il giocatore ha 4900 euro </p> <p>*****</p> <p> [fab] lancia i dadi ed escono: 6 e 6. Dadi doppi!  Si muove da Via![0] a Societa Elettrica[12].  [fab] ha acquistato Societa Elettrica  Il giocatore ha 4850 euro </p> <p>Ritira! *****</p> <p> [fab] lancia i dadi ed escono: 1 e 3.  Si muove da Societa Elettrica[12] a Via Verdi[16].  [fab] ha acquistato Via Verdi  Il giocatore ha 4670 euro </p> <p>*****</p> <p> [Dab] lancia i dadi ed escono: 6 e 5.  Si muove da Via![0] a Via Accademia[11].  [Dab] ha acquistato Via Accademia  Il giocatore ha 4860 euro </p> <p>!</p>

Immagini	<p>Turno 20 *****</p> <p>[cioppo] lancia i dadi ed escono: 5 e 5. Dadi doppi! Si muove da Via Marco Polo[21] a Via Roma[31].</p> <p>Il giocatore ha 4562 euro</p> <p>Ritira! *****</p> <p>[cioppo] lancia i dadi ed escono: 1 e 6. Si muove da Via Roma[31] a Tassa di Lusso[38].</p> <p>Il giocatore ha 4552 euro</p> <p>*****</p> <p>[fab] lancia i dadi ed escono: 3 e 4. Si muove da Stazione Est[35] a Via![0]. fab[4853e] pesca una carta Probabilita' *****</p> <p>Andate avanti sino al Via e ritirate 500e dalla banca. *****</p> <p>Il giocatore ha 5353 euro</p> <p>*****</p> <p>[Dab] lancia i dadi ed escono: 1 e 1. Dadi doppi! Si muove da Piazza università[14] a Via Verdi[16]. Dab paga a fab 18.0 Il giocatore ha 5737 euro</p> <p>Ritira! *****</p> <p>[Dab] lancia i dadi ed escono: 1 e 5. Si muove da Via Verdi[16] a Imprevisti[22]. Dab[5737e] pesca una carta Imprevisti' *****</p> <p>La banca vi sgancia gli interessi del vostro conto corrente: ritirate 125e *****</p> <p>Il giocatore ha 5862 euro</p> <p>Il vincitore e' Dab, con un capitale finale di 5862 euro</p>
----------	--

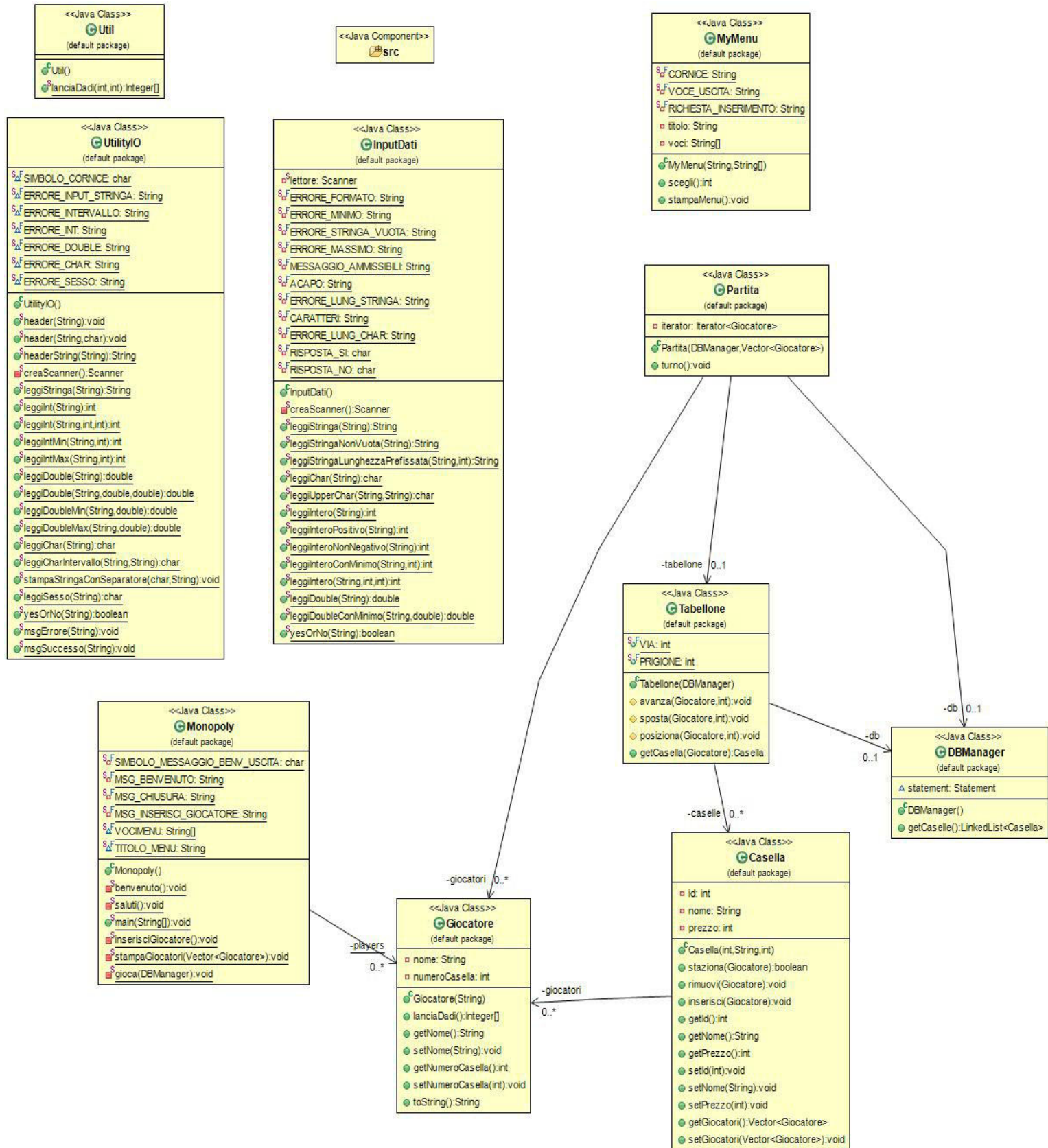
Nome	Rimuovi giocatore
Numero	4
Funzione	L'applicazione permette di rimuovere un giocatore, presentando la lista dei giocatori precedentemente inseriti
Immagini	<p>Digita il numero dell'opzione desiderata &gt; 4</p> <pre> ----- Rimozione Giocatori ----- 1      cioppo[4552e] 2      fab[5371e] 3      Dab[5862e]  0      Esci  Digita il numero dell'opzione desiderata &gt;</pre>
Note	I valori da inserire possono variare rispetto a quelli presentati nell'immagine, a seconda delle opzioni disponibili.

<b>Nome</b>	Esci
<b>Numero</b>	0
<b>Funzione</b>	L'applicazione si chiude.
<b>Immagini</b>	<pre> Digita il numero dell'opzione desiderata &gt; 0 ~~~~~ ~ GRAZIE PER AVERE USATO IL PROGRAMMA - ARRIVEDERCI ~ ~~~~~ </pre>
<b>Note</b>	Per la riesecuzione del programma seguire nuovamente la procedura di esecuzione.

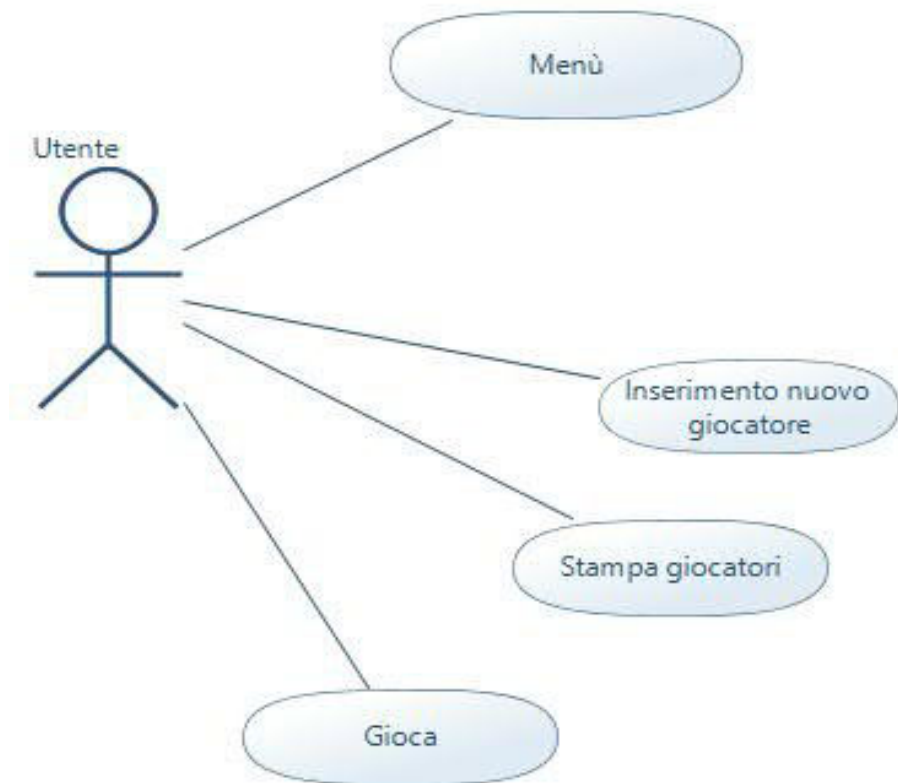


# Release 1

## Diagramma UML delle classi



## Diagramma UML Casi d'uso



## Casi d'uso

Nome	Menu Principale
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'applicazione presenta il menu di scelta</li><li>2. L'utente sceglie l'azione da svolgere</li><li>3. L'utente ha scelto di uscire dall'applicazione</li><li>4. L'applicazione viene terminata.</li></ol>
Scenario Alternativo	<ol style="list-style-type: none"><li>3.a L'utente ha scelto un'operazione da svolgere</li></ol> <p>L'applicazione svolge l'operazione selezionata, torna al punto 1!</p>
Scenario Alternativo	<ol style="list-style-type: none"><li>3.a L'utente digita un input errato</li></ol> <p>Viene visualizzato un messaggio d'errore, Torna al punto 1!</p>

### ***Gestione del menu':***

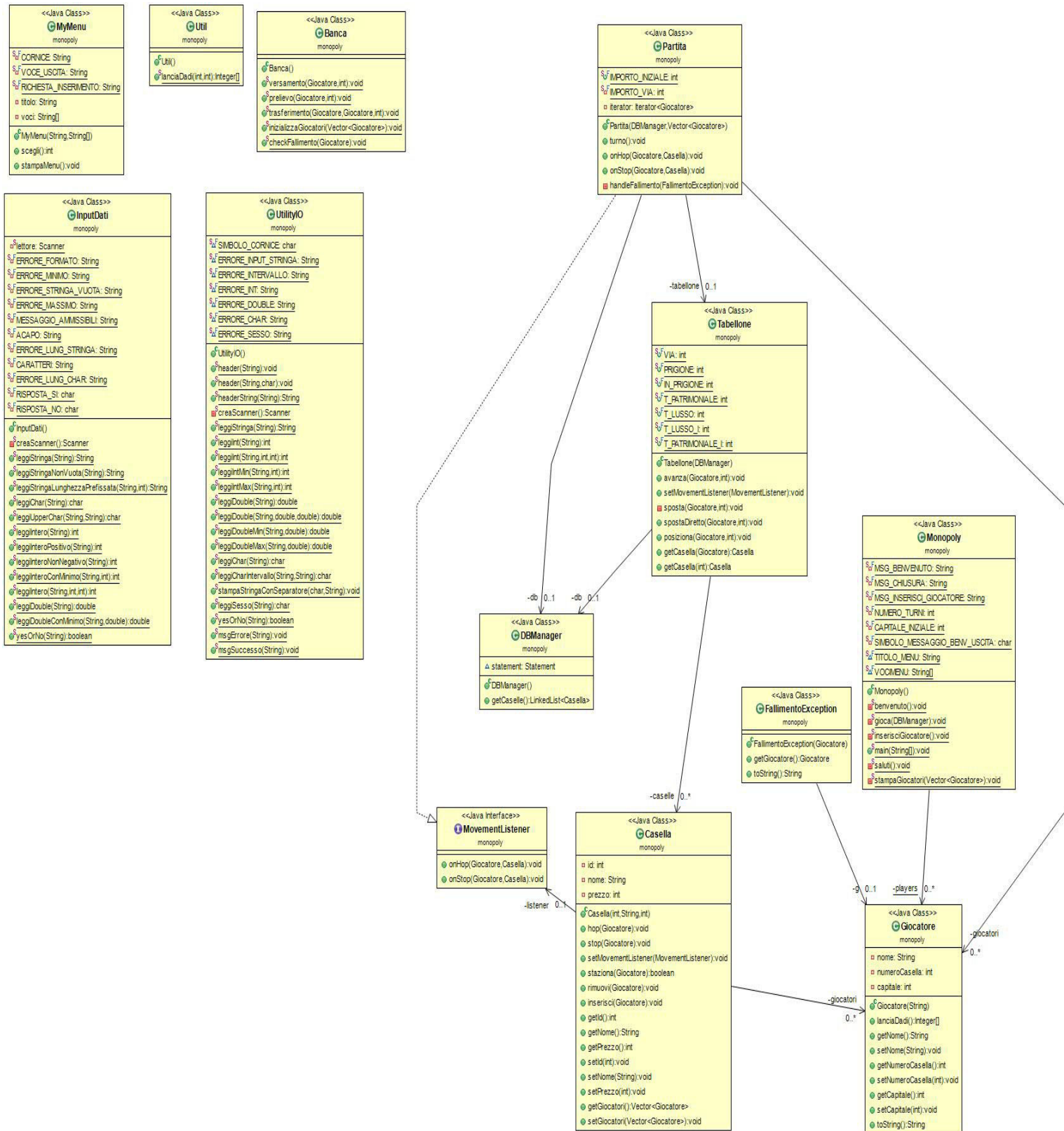
<b>Nome</b>	<b>Inserimento nuovo giocatore</b>
<b>Attore</b>	Utente
<b>Scenario Principale</b>	<ol style="list-style-type: none"><li>1. L'utente sceglie di inserire un nuovo giocatore</li><li>2. Il sistema fa inserire all'utente il nome del giocatore</li><li>3. Il sistema controlla che non ci siano più di 6 giocatori</li></ol> <p>Postcondizione: i nomi dei giocatori vengono memorizzati nel sistema.</p>
<b>Scenario Alternativo</b>	<ol style="list-style-type: none"><li>1.a L'utente decide di inserire un nuovo giocatore</li></ol> <p>Torna al punto 2</p>
<b>Scenario Alternativo</b>	<ol style="list-style-type: none"><li>3.a L'utente inserisce più di 6 giocatori</li></ol> <p>Il sistema rileva che ci sono più di 6 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

<b>Nome</b>	<b>Stampa giocatori</b>
<b>Attore</b>	Utente
<b>Scenario Principale</b>	<ol style="list-style-type: none"><li>1. L'utente sceglie di stampare a video la lista dei giocatori</li><li>2. Il sistema stampa la lista dei giocatori inseriti</li></ol>

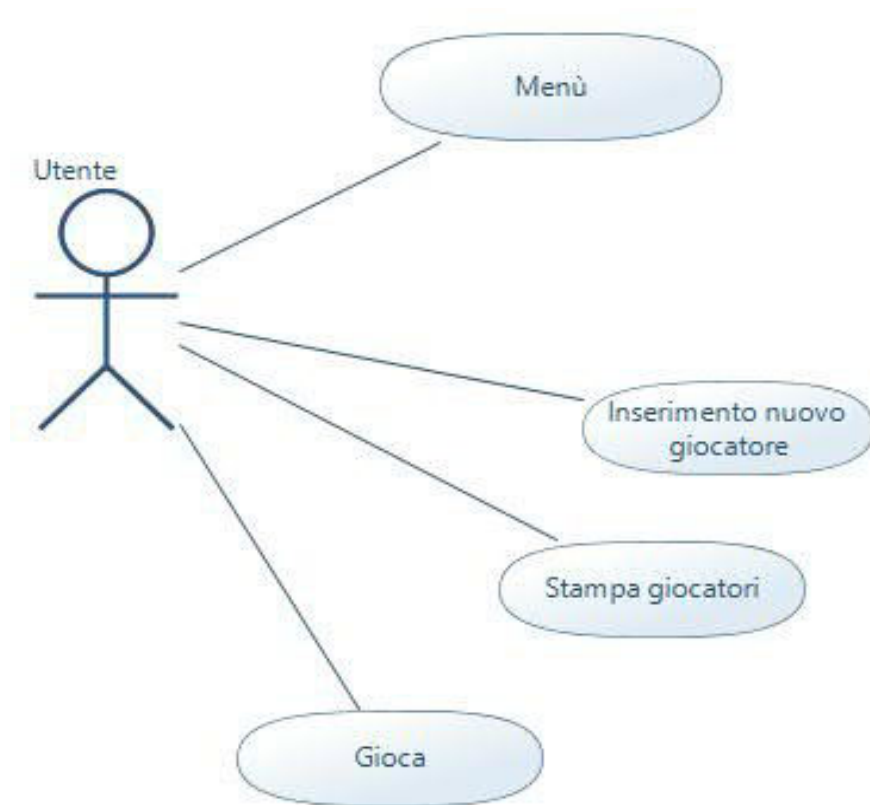
<b>Nome</b>	<b>Gioca</b>
<b>Attore</b>	Utente
<b>Scenario Principale</b>	<ol style="list-style-type: none"><li>1. L'utente sceglie di iniziare la partita</li><li>2. Il sistema controlla che i giocatori siano più di 2</li><li>3. Il sistema svolge la partita</li><li>4. Il sistema termina la partita e stampa a video tutti i turni con i relativi lanci di dadi dei giocatori e i loro movimenti</li></ol>
<b>Scenario Alternativo</b>	<ol style="list-style-type: none"><li>2.a L'utente inserisce meno di 2 giocatori</li></ol> <p>Il sistema rileva che ci sono meno di 2 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

# Release 2

## Diagramma UML delle classi



## Diagramma UML Casi d'uso



## Casi d'uso

Nome	Menu Principale
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'applicazione presenta il menu di scelta</li><li>2. L'utente sceglie l'azione da svolgere</li><li>3. L'utente ha scelto di uscire dall'applicazione</li></ol> <p>L'applicazione viene terminata.</p>
Scenario Alternativo	<p>3.a L'utente ha scelto un'operazione da svolgere</p> <p>L'applicazione svolge l'operazione selezionata</p> <p>Torna al punto 1</p>
Scenario Alternativo	<p>3.a L'utente digita un input errato</p> <p>Viene visualizzato un messaggio d'errore,</p> <p>torna al punto 1</p>

## Gestione del menù:



Nome	Inserimento nuovo giocatore
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"> <li>1. L'utente sceglie di inserire un nuovo giocatore</li> <li>2. Il sistema fa inserire all'utente il nome del giocatore</li> <li>3. Il sistema controlla che non ci siano più di 6 giocatori</li> </ol> <p>Postcondizione: i nomi dei giocatori vengono memorizzati nel sistema.</p>
Scenario Alternativo	<p>1.a L'utente decide di inserire un nuovo giocatore</p> <p>Torna al punto 2</p>
Scenario Alternativo	<p>3.a L'utente inserisce più di 6 giocatori</p> <p>Il sistema rileva che ci sono più di 6 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

---

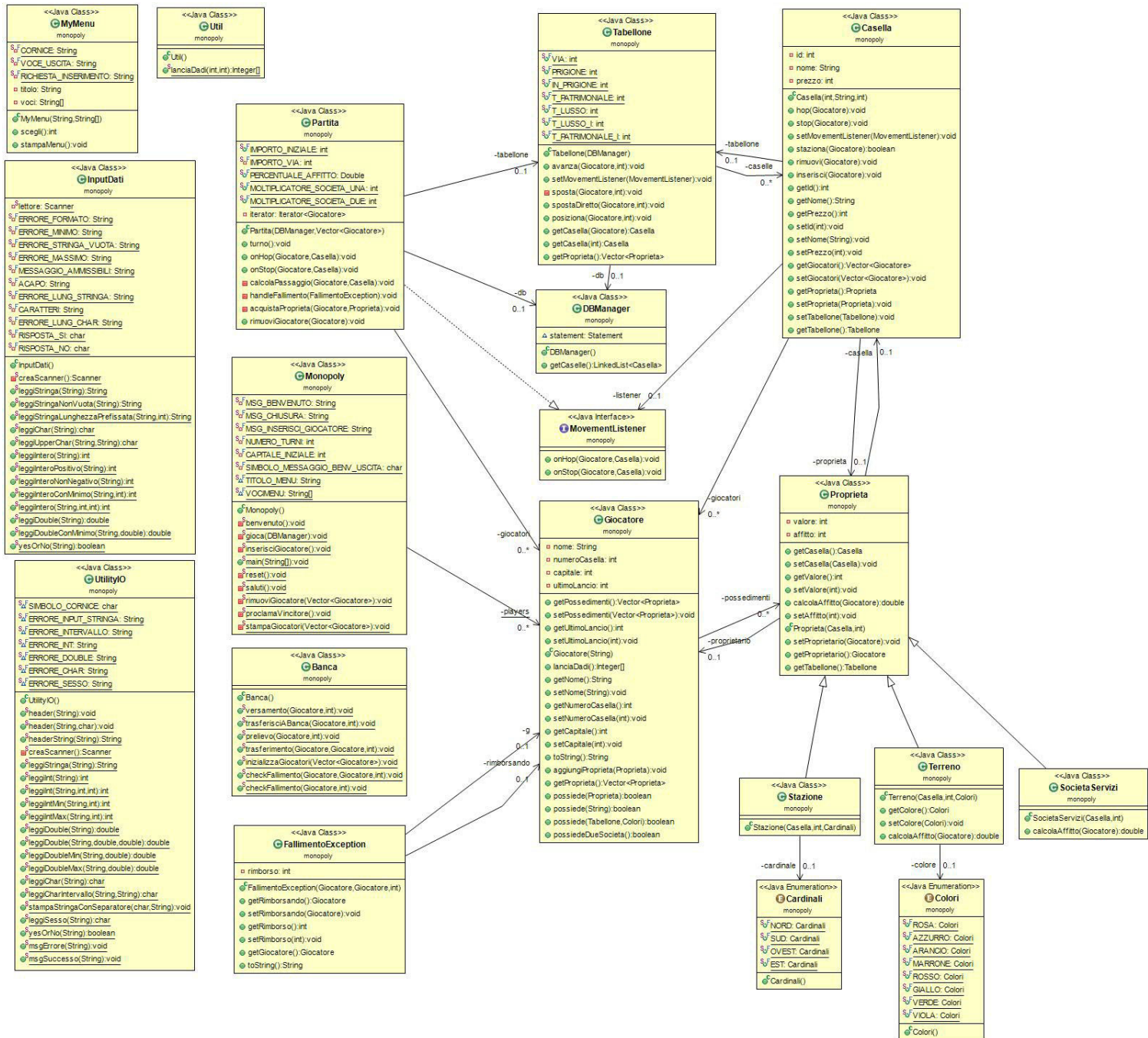
Nome	Stampa giocatori
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"> <li>1. L'utente sceglie di stampare a video la lista dei giocatori</li> <li>2. Il sistema stampa la lista dei giocatori inseriti e il relativo capitale</li> </ol>

Nome	Gioca
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"> <li>1. L'utente sceglie di iniziare la partita</li> <li>2. Il sistema controlla che i giocatori siano più di 2</li> <li>3. Il sistema svolge la partita</li> <li>4. Il sistema termina la partita e stampa a video tutti i turni con i relativi lanci di dadi dei giocatori, i loro movimenti e i loro capitali attuali</li> <li>5. Il sistema dichiara il vincitore</li> </ol>
Scenario Alternativo	<p>2.a L'utente inserisce meno di 2 giocatori</p> <p>Il sistema rileva che ci sono meno di 2 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

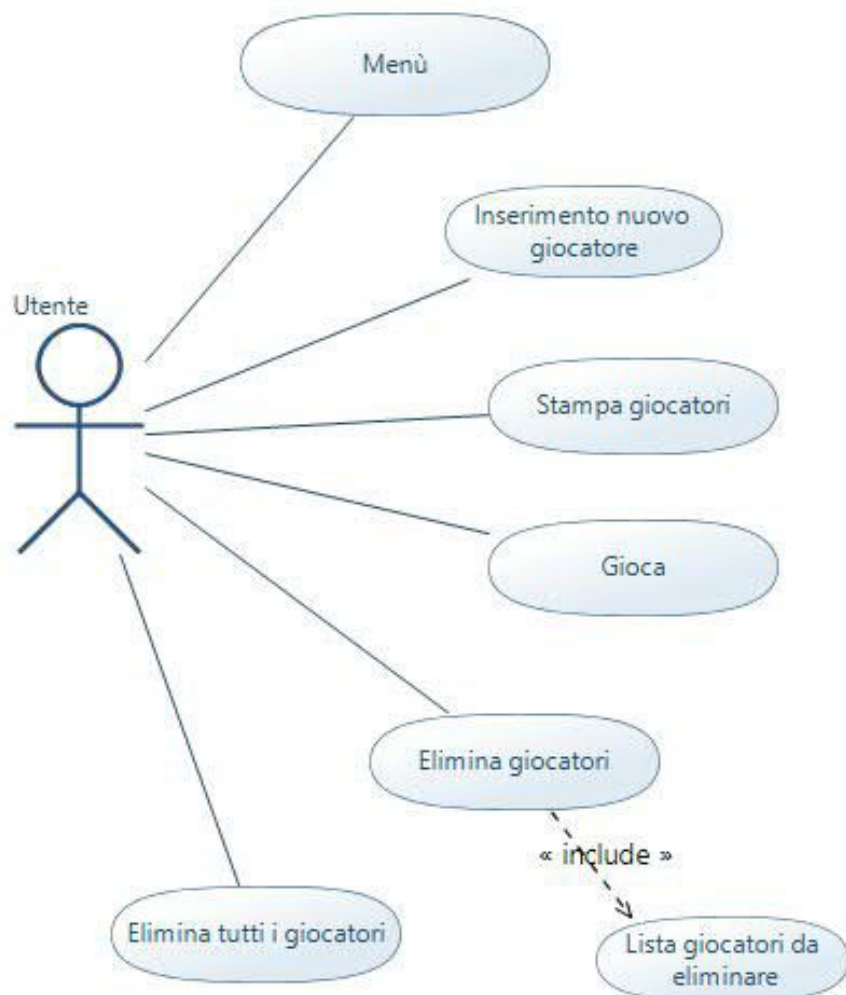
## Release 3

---

## Diagramma UML delle classi



## Diagramma UML casi d'uso



## Casi d'uso

Nome	Menu Principale
Attore	Utente
Scenario Principale	1. L'applicazione presenta il menu di scelta 2. L'utente sceglie l'azione da svolgere 3. L'utente ha scelto di uscire dall'applicazione
Scenario Alternativo	3.a L'utente ha scelto un'operazione da svolgere L'applicazione svolge l'operazione selezionata, torna al punto 1!
Scenario Alternativo	3.a L'utente digita un input errato Viene visualizzato un messaggio d'errore Torna al punto 1!

## Gestione del menù:

Nome	Inserimento nuovo giocatore
Attore	Utente



<b>Scenario Principale</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie di inserire un nuovo giocatore</li> <li>2. Il sistema fa inserire all'utente il nome del giocatore</li> <li>3. Il sistema controlla che non ci siano più di 6 giocatori</li> </ol> <p>Postcondizione: i nomi dei giocatori vengono memorizzati nel sistema.</p>
<b>Scenario Alternativo</b>	<ol style="list-style-type: none"> <li>1.a L'utente decide di inserire un nuovo giocatore</li> </ol> <p>Torna al punto 2</p>
<b>Scenario Alternativo</b>	<ol style="list-style-type: none"> <li>3.a L'utente inserisce più di 6 giocatori</li> </ol> <p>Il sistema rileva che ci sono più di 6 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

<b>Nome</b>	<b>Stampa giocatori</b>
<b>Attore</b>	Utente
<b>Scenario Principale</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie di stampare a video la lista dei giocatori</li> <li>2. Il sistema stampa la lista dei giocatori inseriti</li> </ol>

<b>Nome</b>	<b>Elimina giocatori</b>
<b>Attore</b>	Utente
<b>Scenario Principale</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie di eliminare un giocatore</li> <li>2. &lt;&lt;include&gt;&gt;" Lista dei giocatori da eliminare"</li> <li>3. Il sistema elimina il giocatore scelto dall'utente</li> </ol>

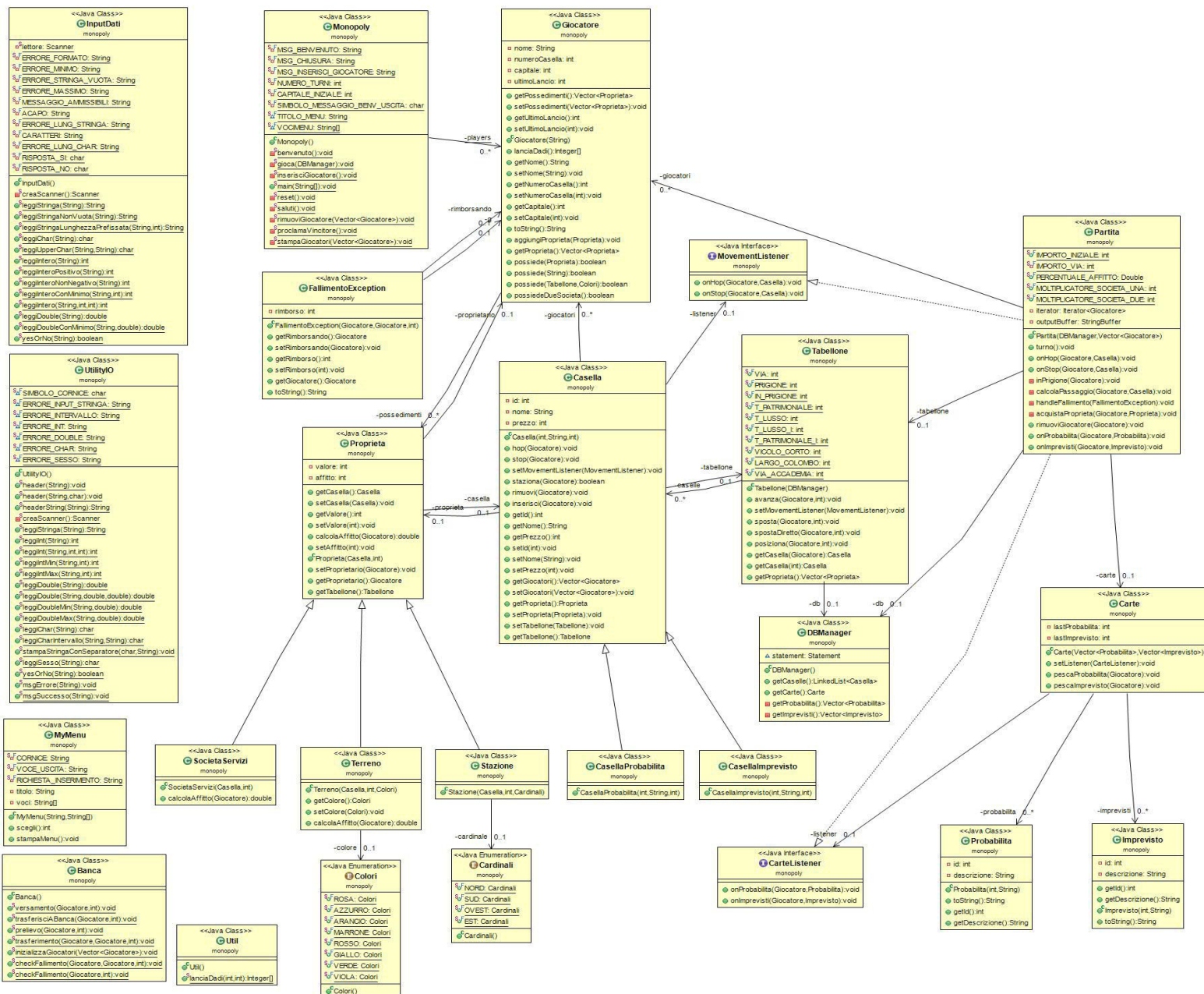
<b>Nome</b>	<b>Elimina tutti i giocatori</b>
<b>Attore</b>	Utente
<b>Scenario Principale</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie di eliminare tutti i giocatori</li> <li>2. Il sistema elimina tutti i giocatori inseriti</li> </ol>

<b>Nome</b>	<b>Gioca</b>
<b>Attore</b>	Utente

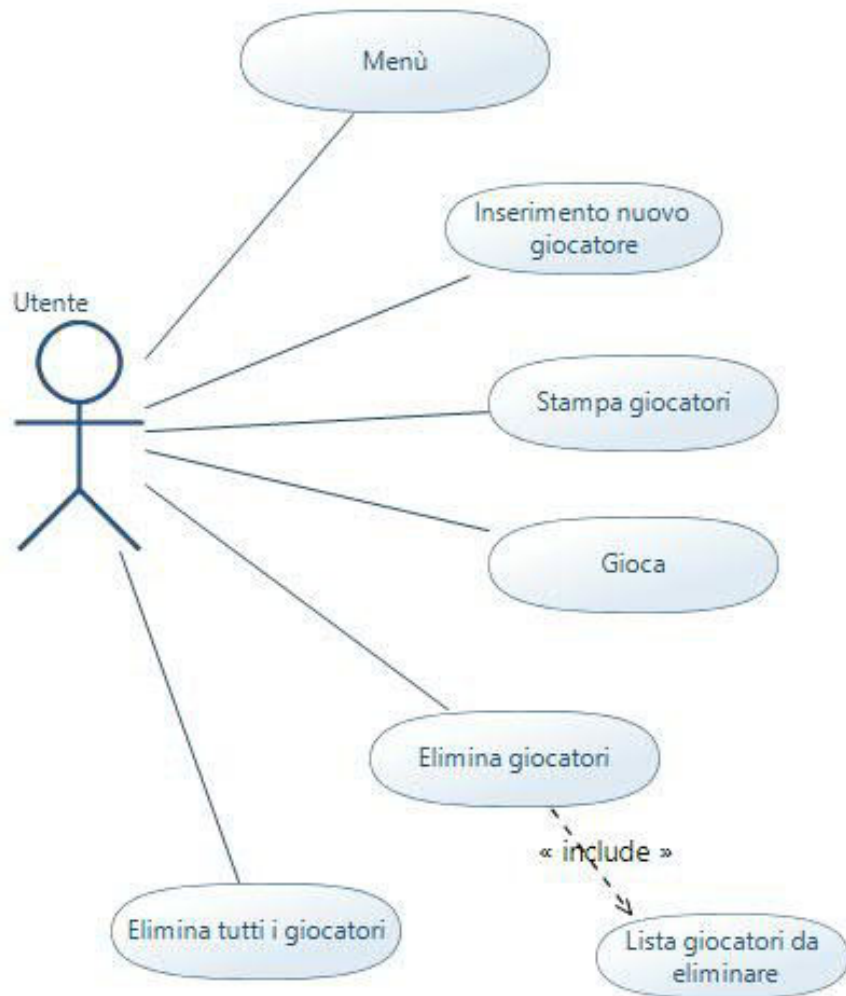
<b>Scenario Principale</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie di iniziare la partita</li> <li>2. Il sistema controlla che i giocatori siano più di 2</li> <li>3. Il sistema svolge la partita</li> <li>4. Il sistema termina la partita e stampa a video tutti i turni con i relativi lanci di dadi dei giocatori, le proprietà acquistate, i loro movimenti, i capitali attuali.</li> <li>5. Il sistema dichiara il fallimento di un giocatore</li> <li>6. Il sistema dichiara il vincitore</li> </ol> <p>Il sistema rileva che un giocatore non ha capitale sufficiente a pagare l'affitto all'avversario e lo dichiara fallito facendolo uscire dalla partita.</p>
<b>Scenario Alternativo</b>	<p>2.a L'utente inserisce meno di 2 giocatori</p> <p>Il sistema rileva che ci sono meno di 2 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

# Release 4

## Diagramma UML delle classi



## Diagramma UML casi d'uso



## Casi d'uso

Nome	Menu Principale
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'applicazione presenta il menu di scelta</li><li>2. L'utente sceglie l'azione da svolgere</li><li>3. L'utente ha scelto di uscire dall'applicazione</li></ol>
Scenario Alternativo	<ol style="list-style-type: none"><li>3.a L'utente ha scelto un'operazione da svolgere</li></ol> <p>L'applicazione svolge l'operazione selezionata, torna al punto 1!</p>
Scenario Alternativo	<ol style="list-style-type: none"><li>3.a L'utente digita un input errato</li></ol> <p>Viene visualizzato un messaggio d'errore Torna al punto 1!</p>

## Gestione del menu':

Nome	Inserimento nuovo giocatore
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'utente sceglie di inserire un nuovo giocatore</li><li>2. Il sistema fa inserire all'utente il nome del giocatore</li><li>3. Il sistema controlla che non ci siano più di 6 giocatori</li></ol> <p>Postcondizione: i nomi dei giocatori vengono memorizzati nel sistema.</p>
Scenario Alternativo	<ol style="list-style-type: none"><li>1.a L'utente decide di inserire un nuovo giocatore</li></ol> <p>Torna al punto 2</p>
Scenario Alternativo	<ol style="list-style-type: none"><li>3.a L'utente inserisce più di 6 giocatori</li></ol> <p>Il sistema rileva che ci sono più di 6 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

Nome	Stampa giocatori
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'utente sceglie di stampare a video la lista dei giocatori</li><li>2. Il sistema stampa la lista dei giocatori inseriti</li></ol>

Nome	Elimina giocatori
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'utente sceglie di eliminare un giocatore</li><li>2. &lt;&lt;include&gt;&gt;" Lista dei giocatori da eliminare"</li><li>3. Il sistema elimina il giocatore scelto dall'utente</li></ol>

Nome	Elimina tutti i giocatori
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'utente sceglie di eliminare tutti i giocatori</li><li>2. Il sistema elimina tutti i giocatori inseriti</li></ol>

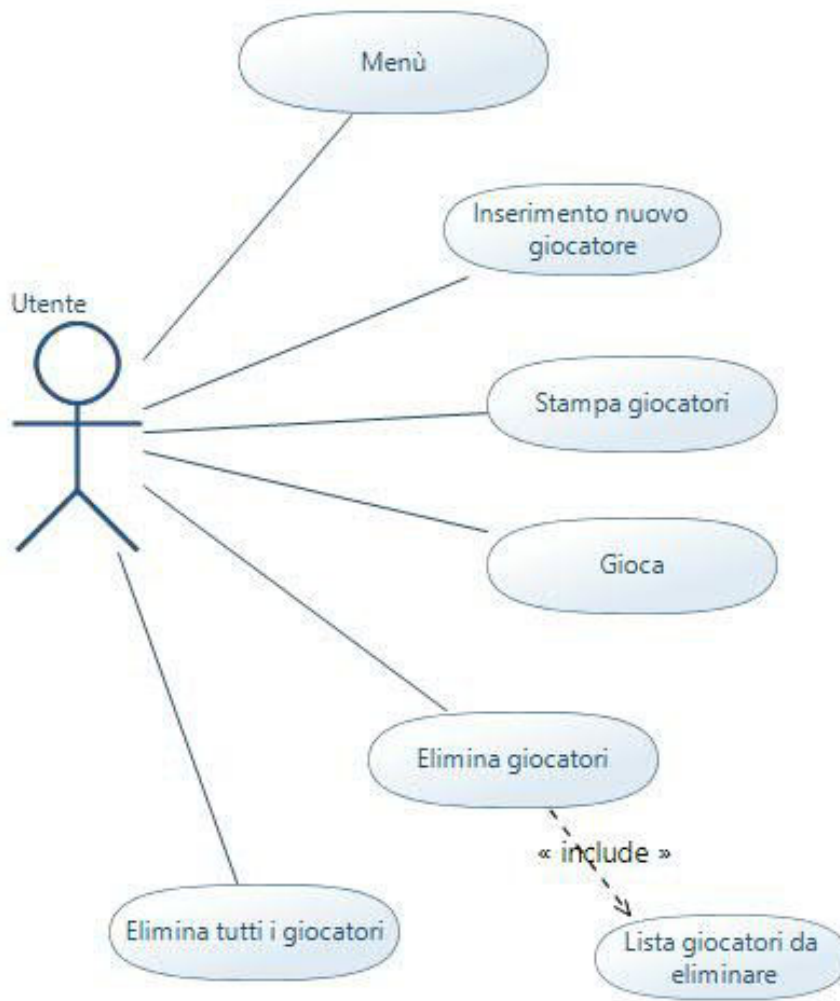
Nome	Gioca
Attore	Utente
<b>Scenario Principale</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie di iniziare la partita</li> <li>2. Il sistema controlla che i giocatori siano più di 2</li> <li>3. Il sistema svolge la partita</li> <li>4. Il sistema termina la partita e stampa a video tutti i turni con i relativi lanci di dadi dei giocatori, le proprietà acquistate, i loro movimenti, i capitali attuali.</li> <li>5. Il sistema dichiara il fallimento di un giocatore</li> <li>6. Il sistema dichiara il vincitore</li> </ol> <p>Il sistema rileva che un giocatore non ha capitale sufficiente a pagare l'affitto all'avversario e lo dichiara fallito facendolo uscire dalla partita.</p>
<b>Scenario Alternativo</b>	<p>2.a L'utente inserisce meno di 2 giocatori</p> <p>Il sistema rileva che ci sono meno di 2 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>



### Diagramma UML delle classi



## Diagramma UML casi d'uso



## Casi d'uso

Nome	Menu Principale
Attore	Utente
Scenario Principale	1. L'applicazione presenta il menu di scelta 2. L'utente sceglie l'azione da svolgere 3. L'utente ha scelto di uscire dall'applicazione
Scenario Alternativo	3.a L'utente ha scelto un'operazione da svolgere  L'applicazione svolge l'operazione selezionata, torna al punto 1!
Scenario Alternativo	3.a L'utente digita un input errato  Viene visualizzato un messaggio d'errore Torna al punto 1!



## Gestione del menu':

Nome	Inserimento nuovo giocatore
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'utente sceglie di inserire un nuovo giocatore</li><li>2. Il sistema fa inserire all'utente il nome del giocatore</li><li>3. Il sistema controlla che non ci siano più di 6 giocatori</li></ol> <p>Postcondizione: i nomi dei giocatori vengono memorizzati nel sistema.</p>
Scenario Alternativo	<ol style="list-style-type: none"><li>1.a L'utente decide di inserire un nuovo giocatore</li></ol> <p>Torna al punto 2</p>
Scenario Alternativo	<ol style="list-style-type: none"><li>3.a L'utente inserisce più di 6 giocatori</li></ol> <p>Il sistema rileva che ci sono più di 6 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

Nome	Stampa giocatori
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'utente sceglie di stampare a video la lista dei giocatori</li><li>2. Il sistema stampa la lista dei giocatori inseriti</li></ol>

Nome	Elimina giocatori
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'utente sceglie di eliminare un giocatore</li><li>2. &lt;&lt;include&gt;&gt;" Lista dei giocatori da eliminare"</li><li>3. Il sistema elimina il giocatore scelto dall'utente</li></ol>

Nome	Elimina tutti i giocatori
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"><li>1. L'utente sceglie di eliminare tutti i giocatori</li><li>2. Il sistema elimina tutti i giocatori inseriti</li></ol>

Nome	Gioca
Attore	Utente
<b>Scenario Principale</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie di iniziare la partita</li> <li>2. Il sistema controlla che i giocatori siano più di 2</li> <li>3. Il sistema svolge la partita</li> <li>4. Il sistema termina la partita e stampa a video tutti i turni con i relativi lanci di dadi dei giocatori, le proprietà acquistate, i loro movimenti, i capitali attuali.</li> <li>5. Il sistema dichiara il fallimento di un giocatore</li> <li>6. Il sistema dichiara il vincitore</li> </ol> <p>Il sistema rileva che un giocatore non ha capitale sufficiente a pagare l'affitto all'avversario e lo dichiara fallito facendolo uscire dalla partita.</p>
<b>Scenario Alternativo</b>	<p>2.a L'utente inserisce meno di 2 giocatori</p> <p>Il sistema rileva che ci sono meno di 2 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

# Testing

---

Test Suite	1
Software Testato	Menu Principale.
Versione	1
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Risultato	Il Menu Principale risponde correttamente ai comandi dell'utente.
Autore	Cordioli Francesco.
Data	09/03/2015

Test Case	1
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Ambiente	<i>Driver</i> Main <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	0
Uscite attese	Uscita dal programma con relativo messaggio di uscita.
Uscite effettive	Le stesse.
Risultato'	<b>PASS!</b>  Il menu principale risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	2
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Ambiente	<i>Driver</i> Main <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	1

<b>Uscite attese</b>	Inserimento e salvataggio di un nuovo giocatore.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il menu principale risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>3</b>
<b>Obiettivo</b>	Verificare il corretto funzionamento del Menu Principale.
<b>Ambiente</b>	<i>Driver</i> Main <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	2
<b>Uscite attese</b>	Visualizzazione dei giocatori inseriti dall'utente.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il menu principale risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>4</b>
<b>Obiettivo</b>	Verificare il corretto funzionamento del Menu Principale.
<b>Ambiente</b>	<i>Driver</i> Main <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	3
<b>Uscite attese</b>	Visualizzazione della partita giocata con il numero della faccia dei dadi tirati e spostamento del giocatore.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il menu principale risponde correttamente ai comandi dell'utente.

Osservazioni	
--------------	--

<b>Test Case</b>	<b>5</b>
<b>Obiettivo</b>	Verificare il corretto funzionamento del Menu Principale.
<b>Ambiente</b>	<i>Driver:</i> Main <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	4
<b>Uscite attese</b>	Messaggio di errore e ritorno al menu principale.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	E' stato dato in ingresso un'opzione del menù che non esiste.

<b>Test Suite</b>	<b>2</b>
<b>Software Testato</b>	Inserimento dei giocatori.
<b>Versione</b>	1
<b>Obiettivo</b>	Verificare il corretto inserimento dei giocatori e il numero massimo di essi.
<b>Risultato</b>	L'inserimento risponde correttamente ai comandi dell'utente.
<b>Autore</b>	Cordioli Francesco.
<b>Data</b>	09/03/2015

<b>Test Case</b>	<b>1</b>
<b>Obiettivo</b>	Verificare il corretto inserimento dei dati.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	"Dab".
<b>Uscite attese</b>	Salvataggio del nome.
<b>Uscite effettive</b>	Le stesse.

<b>Risultato'</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	Se vengono inseriti più di 6 giocatori viene stampato a video un messaggio che informa l'utente che è stato raggiunto il numero massimo di giocatori.

<b>Test Case</b>	<b>2</b>
<b>Obiettivo</b>	Verificare il corretto inserimento dei dati
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Messaggio d'errore.
<b>Uscite effettive</b>	Salvataggio del nome.
<b>Risultato'</b>	<b>FAIL!</b> Il programma non risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	Manca il controllo sul mancato inserimento del nome del giocatore.

<b>Test Case</b>	<b>3</b>
<b>Obiettivo</b>	Verificare il corretto inserimento dei dati
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Messaggio d'errore.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	Sistemato il controllo sull'inserimento del nome.

<b>Test Suite</b>	<b>3</b>
<b>Software Testato</b>	Visualizzazione dei giocatori.
<b>Versione</b>	1
<b>Obiettivo</b>	Verificare la corretta visualizzazione della lista dei giocatori inseriti dall'utente.
<b>Risultato</b>	La visualizzazione risponde correttamente ai comandi dell'utente.
<b>Autore</b>	Cordioli Francesco.
<b>Data</b>	09/03/2015

<b>Test Case</b>	<b>1</b>
<b>Obiettivo</b>	Verificare la corretta visualizzazione della lista dei giocatori.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Lista dei nomi.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Suite</b>	<b>4</b>
<b>Software Testato</b>	Gioco-Partita.
<b>Versione</b>	1
<b>Obiettivo</b>	Verificare il corretto funzionamento del gioco.
<b>Risultato</b>	Il gioco funziona come da consegna.
<b>Autore</b>	Cordioli Francesco.
<b>Data</b>	09/03/2015

Test Case	1
Obiettivo	Verificare il corretto funzionamento del gioco.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Lista di 20 turni con varie descrizioni di dadi e posizioni aggiornate.
Uscite effettive	Le stesse.
Risultato'	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	Il programma da messaggio di errore se sono stati inseriti meno di 2 giocatori.

Test Case	2
Obiettivo	Verificare che tutti i giocatori inizino la partita dalla casella VIA!
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	I giocatori si trovano tutti sulla casella VIA!
Uscite effettive	Le stesse.
Risultato'	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	3
Obiettivo	Verificare che dopo 3 volte che il giocatore ottiene lo stesso punteggio per entrambi i dadi finisce in prigione.



<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Il giocatore dopo 3 tiri uguali finisce in prigione.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>4</b>
<b>Obiettivo</b>	Verificare che una volta che il giocatore arriva alla casella 40 riparte dalla casella VIA!
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Alla conclusione di ogni giro il conteggio della posizione del giocatore riparte da 0.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.

<b>Test Case</b>	<b>5</b>
<b>Obiettivo</b>	Verificare che una partita termina dopo 20 turni.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Termine della partita.

<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Suite</b>	<b>5</b>
<b>Software Testato</b>	Gioco-partita con l'interazione della banca.
<b>Versione</b>	1
<b>Obiettivo</b>	Verificare il corretto funzionamento del Menu Principale.
<b>Risultato</b>	Il risultato finale corrisponde alla consegna data.
<b>Autore</b>	Cordioli Francesco.
<b>Data</b>	10/03/2015

<b>Test Case</b>	<b>1</b>
<b>Obiettivo</b>	Verificare che ad ogni giocatore sia assegnata la cifra di 5000€.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Ogni giocatore ha un capitale di 5000€.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>2</b>
<b>Obiettivo</b>	Verificare che ad ogni passaggio dal via il giocatore riceva 500€.

<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Ogni utente riceve 500€ dalla banca.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>3</b>
<b>Obiettivo</b>	Verificare che dalla casella 30 ("IN PRIGIONE") venga spostato alla casella 10 ("PRIGIONE/TRANSITO") senza ricevere i 500€ dalla banca per il passaggio dal via.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Passaggio dalla casella 30 alla casella 10 e i soldi devono rimanere gli stessi.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

Test Case	4
Obiettivo	Verificare il pagamento di 250€ alla banca per aver sostato sulla casella TASSA PATRIMONIALE.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore ha 250€ in meno.
Uscite effettive	Le stesse.
Risultato	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	5
Obiettivo	Verificare il pagamento di 10€ alla banca per aver sostato sulla casella TASSA DEL LUSSO.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore ha 10€ in meno.
Uscite effettive	Le stesse.
Risultato	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	6
Obiettivo	Verificare il fallimento di un giocatore e l'uscita di esso dalla partita.

<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Il giocatore esce dalla partita.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>7</b>
<b>Obiettivo</b>	Verificare il vincitore della partita giocata.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Messaggio a video con il nome del vincitore.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	Il giocatore ha vinto perché o tutti gli altri giocatori hanno fallito oppure è quello più ricco.

<b>Test Suite</b>	<b>6</b>
<b>Software Testato</b>	Menu Principale.
<b>Versione</b>	1
<b>Obiettivo</b>	Verificare il corretto funzionamento delle due voci aggiunte al menù principale.
<b>Risultato</b>	Il Menu Principale risponde correttamente ai comandi dell'utente.

<b>Autore</b>	Falletti Davide.
<b>Data</b>	20/03/2015

<b>Test Case</b>	1
<b>Obiettivo</b>	Verificare il corretto funzionamento del Menu Principale.
<b>Ambiente</b>	<i>Driver Main Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	4
<b>Uscite attese</b>	Visualizzazione di una lista di giocatori per l'eliminazione.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b> Il menu principale risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	Visualizzo una lista di giocatori da eliminare; per poter scegliere quale eliminare premo il numero della lista corrispondente al giocatore.

<b>Test Case</b>	2
<b>Obiettivo</b>	Verificare il corretto funzionamento del Menu Principale.
<b>Ambiente</b>	<i>Driver Main Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	5
<b>Uscite attese</b>	Eliminazione di tutti i giocatori.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b> Il menu principale risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Suite</b>	7
<b>Software Testato</b>	Gioco-partita con l'interazione della banca e con il valore su ogni casella.
<b>Versione</b>	1
<b>Obiettivo</b>	Verificare il corretto funzionamento del Menu Principale.
<b>Risultato</b>	Il risultato finale corrisponde alla consegna data.
<b>Autore</b>	Falletti Davide.
<b>Data</b>	20/03/2015

<b>Test Case</b>	1
<b>Obiettivo</b>	Verificare che ogni casella abbia un valore in euro.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Ogni casella ha il suo costo.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato</b>	PASS!  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	2
<b>Obiettivo</b>	Verificare che quando un giocatore arriva su una casella che non è ancora stata acquistata, la acquista.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.

<b>Uscite attese</b>	Il giocatore acquista automaticamente la proprietà.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>3</b>
<b>Obiettivo</b>	Verificare che il giocare acquisti la proprietà solo se ha denaro sufficiente.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Messaggio di fondi insufficienti e non avviene l'acquisto della proprietà.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>4</b>
<b>Obiettivo</b>	Verificare che quando un giocatore arriva su un terreno o una stazione di proprietà di un altro giocatore, deve pagare a quest'ultimo il 10% del valore della casella.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Pagamento del 10% della casella all'avversario.
<b>Uscite effettive</b>	Le stesse.



<b>Risultato</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>5</b>
<b>Obiettivo</b>	Verificare che quando un giocatore arriva su una società di proprietà di un altro giocatore, l'affitto deve essere 4 volte il numero ottenuto con i dadi.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	L'affitto pagato (in €) all'avversario è dato dal valore dei dadi moltiplicato per 4.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	Questo avviene se l'avversario possiede una sola società di servizi.

<b>Test Case</b>	<b>6</b>
<b>Obiettivo</b>	Verificare che quando un giocatore arriva su una società di proprietà di un altro giocatore, l'affitto deve essere 10 volte il numero ottenuto con i dadi.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	L'affitto pagato (in €) all'avversario è dato dal valore dei dadi moltiplicato per 10.
<b>Uscite effettive</b>	Le stesse.

<b>Risultato</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	Questo avviene se l'avversario possiede entrambe le società di servizi.

<b>Test Case</b>	<b>7</b>
<b>Obiettivo</b>	Verificare che se il giocatore possiede tutti i terreni di un certo colore, l'affitto raddoppia.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	L'affitto pagato all'avversario è doppio.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>8</b>
<b>Obiettivo</b>	Verificare il fallimento di un giocatore.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Il giocatore esce dal gioco.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	Questo avviene se un giocatore non possiede un capitale sufficiente per pagare per intero l'affitto dovuto a un altro giocatore.

Test Case	9
<b>Obiettivo</b>	Verificare che se un giocatore si sposta su una casella di un altro giocatore ma non ha i soldi per pagare l'affitto, fallisce dando il suo capitale al giocatore che possiede la proprietà e quest'ultimo riceve la differenza del valore dovuto dalla banca.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	L'avversario riceve i soldi dal giocatore che ha fallito e riceve il resto dei soldi dalla banca.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

Test Case	10
<b>Obiettivo</b>	Verificare che le proprietà del giocatore fallito ritornino acquistabili.
<b>Ambiente</b>	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Le proprietà tornano acquistabili.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato</b>	<b>PASS!</b>  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Suite</b>	8
<b>Software Testato</b>	Gioco-partita con l'aggiunta delle PROBABILITA' e degli IMPREVISTI.
<b>Versione</b>	1
<b>Obiettivo</b>	Verificare il corretto funzionamento del Menu Principale.
<b>Risultato</b>	Il risultato finale corrisponde alla consegna data.
<b>Autore</b>	Falletti Davide.
<b>Data</b>	27/03/2015

<b>Test Case</b>	1
<b>Obiettivo</b>	Verificare che all'inizio di ogni partita ciascuno dei due mazzi (IMPREVISTI e PROBABILITA') venga mescolato casualmente.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Mazzo mescolato casualmente.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	2
<b>Obiettivo</b>	Verificare che il giocatore quando arriva su una casella di probabilità o imprevisti, peschi la carta in cima al mazzo.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Il giocatore pesca la prima carta del mazzo.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.

Osservazioni	
--------------	--

Test Case	3
Obiettivo	Verificare che il giocatore esegua l'operazione scritta sulla carta di probabilità o imprevisti.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore si comporterà in modo diverso in base all'operazione scritta sulla carta.
Uscite effettive	Le stesse.
Risultato'	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	4
Obiettivo	Verificare che il giocatore depositi la carta in fondo al mazzo.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore mette la carta in fondo al mazzo.
Uscite effettive	Le stesse.
Risultato'	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	5
Obiettivo	Verificare il fallimento di un giocatore.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>

<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Il giocatore fallisce ed esce dalla partita.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	PASS!  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	Un giocatore fallisce quando non possiede un capitale sufficiente da eseguire l'azione scritta sulla carta di probabilità o di imprevisti.

<b>Test Case</b>	<b>6</b>
<b>Obiettivo</b>	Verificare che se un giocatore fallisce, l'eventuale resto che dovrebbe dare all'avversario venga fornito dalla banca.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	L'avversario riceve una certa quantità di denaro dalla banca.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	PASS!  Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	L'eventuale resto è dovuto dal fatto che il giocatore non ha soldi per eseguire l'azione scritta sulla carta di probabilità o imprevisti e fallendo cede tutto il suo capitale all'avversario a cui è dovuto.

<b>Test Suite</b>	<b>9</b>
<b>Software Testato</b>	Gioco-partita con l'aggiunta di alcune regole per la gestione della PRIGIONE.
<b>Versione</b>	1
<b>Obiettivo</b>	Verificare il corretto funzionamento del Menu Principale.
<b>Risultato</b>	Il risultato finale corrisponde alla consegna data.
<b>Autore</b>	Cordioli Francesco.
<b>Data</b>	03/04/2015

Test Case	1
Obiettivo	Verificare che il giocatore finisca in prigione quando pesca una carta di PROBABILITA' O IMPREVISITI che gli indica di andare in prigione.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore entra in prigione.
Uscite effettive	Le stesse.
Risultato'	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	2
Obiettivo	Verificare che il giocatore finisca in prigione quando, tirando i dadi, esce per tre volte dadi doppi.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore finisce in prigione.
Uscite effettive	Le stesse.
Risultato'	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	3
Obiettivo	Verificare che, al turno successivo, prima di tirare i dadi, il giocatore paghi 50€ alla banca per uscire di prigione.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>

<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Il giocatore esce di prigione e la banca ha 50€ in più.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>4</b>
<b>Obiettivo</b>	Verificare che se un giocatore è in prigione e non possiede un capitale sufficiente per uscirne, fallisce.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Il giocatore fallisce.
<b>Uscite effettive</b>	Le stesse.
<b>Risultato'</b>	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
<b>Osservazioni</b>	

<b>Test Case</b>	<b>5</b>
<b>Obiettivo</b>	Verificare che tutte le proprietà del giocatore fallito ritornino acquistabili.
<b>Ambiente</b>	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
<b>Dati in ingresso</b>	Nessuno.
<b>Uscite attese</b>	Le proprietà sono riacquistabili.
<b>Uscite effettive</b>	Le stesse.



Risultato'	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	6
Obiettivo	Verificare che tutto il capitale del giocatore fallito venga ceduto alla banca.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	La banca ha in più il denaro del giocatore fallito.
Uscite effettive	Le stesse.
Risultato'	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	7
Obiettivo	Verificare che il giocatore fallito esca dalla partita.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore non c'è più nella partita.
Uscite effettive	Le stesse.
Risultato'	<b>PASS!</b> Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	