

# QUESTIONARIO ESAMI SAS

## 1. Domanda #1

In UP, la disciplina dei requisiti ha l'obiettivo di produrre una lista dei requisiti, capire il contesto del sistema, catturare i requisiti funzionali e i requisiti non-funzionali

TRUE



CORRETTO !

## 2. Domanda #2

Il modello di dominio include la definizione di oggetti, associazioni e attributi di classi software

FALSE



CORRETTO !

## 3. Domanda #3

Il modello di dominio è un dizionario visuale delle classi concettuali

TRUE



CORRETTO !

## 4. Domanda #4

Il modello di dominio è una rappresentazione testuale delle classi concettuali, oggetti reali del dominio

FALSE



CORRETTO !

## 5. Domanda #5

Il modello di dominio non è parte della disciplina dei requisiti

TRUE



CORRETTO !

### Motivo:

fa parte della disciplina di modellazione del business

## 6. Domanda #6

Nelle associazioni nel modello di dominio la direzione di lettura va sempre specificata

FALSE



CORRETTO !

## 7. Domanda #7

L'associazione nel modello di dominio è per sua natura unidirezionale

FALSE



CORRETTO !

## 8. Domanda #8

L'associazione nel modello di dominio rappresenta una relazione significativa tra classi

TRUE



CORRETTO !

**9. Domanda #9**

L'associazione nel modello di dominio rappresenta un valore logico degli oggetti di una classe

 FALSE**CORRETTO !****10. Domanda #10**

L'associazione nel modello di dominio rappresenta un'insieme di n-tuple di oggetti delle classi

 TRUE**CORRETTO !****11. Domanda #11**

Il modello di dominio è una rappresentazione testuale delle classi concettuali, oggetti reali del dominio

 FALSE**CORRETTO !****12. Domanda #12**

Il modello di dominio è un dizionario visuale delle classi concettuali

 TRUE**CORRETTO !****13. Domanda #13**

il MODELLO DI DOMINIO: Insieme di diagrammi di classi UML che includono associazioni tra classi software

 FALSE**CORRETTO !****14. Domanda #14**

il MODELLO DI DOMINIO: Rappresentazione grafica degli oggetti software

 FALSE**CORRETTO !****15. Domanda #15**

il MODELLO DI DOMINIO: Rappresentazione visuale delle classi concettuali

 TRUE**CORRETTO !****16. Domanda #16**

il MODELLO DI DOMINIO: Insieme di diagrammi di classi UML che includono associazioni tra classi concettuali

 TRUE**CORRETTO !****17. Domanda #17**

il MODELLO DI DOMINIO: Rappresentazione visuale delle classi JAVA

 FALSE**CORRETTO !**

## 18. Domanda #18

il MODELLO DI DOMINIO: Insieme di diagrammi di classi UML che includono le responsabilità di fare

FALSE



CORRETTO !

## 19. Domanda #19

Un caso d'uso rappresenta una maniera di utilizzare il sistema da parte di un utente per raggiungere un suo obiettivo

TRUE



CORRETTO !

## 20. Domanda #20

La narrativa di un caso d'uso viene espressa a livello delle intenzioni dell'utente e della responsabilità del sistema

TRUE



CORRETTO !

## 21. Domanda #21

La narrativa di un caso d'uso viene espressa a livello delle azioni concrete dell'utente e delle responsabilità del sistema

FALSE



CORRETTO !

## 22. Domanda #22

Un caso d'uso è sempre completato (implementato) in una sola iterazione

FALSE



CORRETTO !

## 23. Domanda #23

Un caso d'uso rappresenta l'insieme di funzionalità di un sistema

FALSE



CORRETTO !

## 24. Domanda #24

I requisiti funzionali non catturati dai casi d'uso vengono descritti nelle specifiche supplementari

FALSE



CORRETTO !

## 25. Domanda #25

L'analisi linguistica dei casi d'uso nel formato dettagliato è una fonte di ispirazione per la costruzione del modello di dominio

TRUE



CORRETTO !

## 26. Domanda #26

Il modello di dominio riporta i concetti significativi relativi ai casi d'uso

TRUE

CORRETTO !

## 27. Domanda #27

L'analisi linguistica dei casi d'uso nel formato dettagliato è una fonte di ispirazione per la costruzione del modello di dominio

 TRUE

CORRETTO !

## 28. Domanda #28

I requisiti non funzionali sono descritti completamente dai casi d'uso

 FALSE

CORRETTO !

## 29. Domanda #29

I contratti sono considerati parte del modello dei casi d'uso poiché forniscono maggiori dettagli dell'analisi, sull'effetto delle operazioni di sistema implicate dai casi d'uso

 TRUE

CORRETTO !

## 30. Domanda #30

I CASI D'USO Sono una collezione di soli scenari di successo che descrivono un attore che usa il sistema per raggiungere un obiettivo specifico

 FALSE

CORRETTO !

## 31. Domanda #31

I casi d'uso sono descrizioni testuali di scenari di uso interessanti del sistema software che si deve realizzare

 TRUE

CORRETTO !

## 32. Domanda #32

I casi d'uso servono a catturare i requisiti funzionali

 TRUE

CORRETTO !

## 33. Domanda #33

In UP i casi d'uso vengono utilizzati per catturare i requisiti funzionali nella disciplina dei requisiti

 TRUE

CORRETTO !

## 34. Domanda #34

I casi d'uso sono caratteristiche del sistema

 FALSE

CORRETTO !

### 35. Domanda #35

I casi d'uso sono utilizzati per pianificare le iterazioni

TRUE



CORRETTO !

### 36. Domanda #36

In UP i casi d'uso sono descritti mediante il linguaggio UML

FALSE



CORRETTO !

### 37. Domanda #37

DURANTE LA FASE DI ELABORAZIONE Si scrivono circa il 10% dei casi d'uso tra i più critici in formato dettagliato utilizzando template appositi

FALSE



CORRETTO !

### 38. Domanda #38

In percentuale sul totale di casi d'uso complessivi di un progetto, quanti sono quelli che dovrebbero essere realizzati entro la conclusione della fase di ideazione??

5%



CORRETTO !

### 39. Domanda #39

I casi d'uso mettono in risalto gli obiettivi degli utenti

TRUE



CORRETTO !

### 40. Domanda #40

I casi d'uso sono utilizzati per la scoperta e la definizione dei requisiti non funzionali

FALSE



CORRETTO !

### 41. Domanda #41

I casi d'uso sono utilizzati per la scoperta e la definizione dei requisiti funzionali

TRUE



CORRETTO !

### 42. Domanda #42

I casi d'uso mettono in risalto gli obiettivi del sistema

FALSE



CORRETTO !

### 43. Domanda #43

I casi d'uso sono utilizzati solo nelle prime iterazioni di sviluppo

FALSE



CORRETTO !

#### 44. Domanda #44

Il principale input per la redazione dei contratti sono le operazioni di sistema e il modello di dominio

TRUE



CORRETTO !

#### 45. Domanda #45

Le post-condizioni descrivono i cambiamenti di stato degli oggetti nel modello di dominio dopo il completamento dell'operazione

TRUE



CORRETTO !

#### 46. Domanda #46

I contratti servono come input per il modello di dominio

FALSE



CORRETTO !

#### 47. Domanda #47

La responsabilità sono assegnate ai concetti definiti nel modello di dominio

FALSE



CORRETTO !

#### 48. Domanda #48

I contratti delle operazioni di sistema usano pre e post- condizioni per descrivere nel dettaglio i cambiamenti agli oggetti (concettuali) in un modello di dominio

TRUE



CORRETTO !

#### 49. Domanda #49

I contratti servono come input per il modello di dominio

FALSE



CORRETTO !

#### 50. Domanda #50

Le pre-condizioni descrivono i cambiamenti di stato degli oggetti nel modello di dominio dopo il completamento dell'operazione

FALSE



CORRETTO !

#### 51. Domanda #51

La composizione di oggetti è definita staticamente attraverso la specifica delle classi e delle associazioni

FALSE



CORRETTO !

#### 52. Domanda #52

Il meccanismo di specializzazione per il riuso del codice è detto white-box

TRUE



CORRETTO !

### 53. Domanda #53

La modifica dell'interfaccia di una classe ha in generale un basso impatto nelle sottoclassi

 FALSE**CORRETTO !**

### 54. Domanda #54

I pattern GRASP sono espressi in termini di responsabilità, ruoli e collaborazioni

 TRUE**CORRETTO !**

### 55. Domanda #55

La decisione sull'assegnazione delle responsabilità precede sempre la codifica

 FALSE**CORRETTO !**

### 56. Domanda #56

L'utilizzo dei pattern GRASP è svolto all'interno della disciplina dei requisiti di UP

 FALSE**CORRETTO !**

### 57. Domanda #57

Le responsabilità sono assegnate alle classi durante la OOD

 TRUE**CORRETTO !**

### 58. Domanda #58

Il meccanismo di delega è preferibile al meccanismo di specializzazione per il riuso del codice

 TRUE**CORRETTO !**

### 59. Domanda #59

L'ereditarietà rispetta l'incapsulamento

 FALSE**CORRETTO !**

### 60. Domanda #60

La composizione di oggetti è un meccanismo di riuso del codice detto black-box

 TRUE**CORRETTO !**

### 61. Domanda #61

Un metodo di una classe software rappresenta una responsabilità "a fare" per le istanze di tale classe

 TRUE**CORRETTO !**

## 62. Domanda #62

DURANTE LA FASE DI ELABORAZIONE Non vengono effettuati test al codice di sviluppo

FALSE



CORRETTO !

## 63. Domanda #63

DURANTE LA FASE DI ELABORAZIONE Non si sviluppa codice

FALSE



CORRETTO !

## 64. Domanda #64

DURANTE LA FASE DI ELABORAZIONE Viene scoperta e stabilizzata la maggior parte dei requisiti

TRUE



CORRETTO !

## 65. Domanda #65

DURANTE LA FASE DI ELABORAZIONE Si realizza uno studio economico per stabilire l'ordine di grandezza del progetto e dei costi

FALSE



CORRETTO !

## 66. Domanda #66

DURANTE LA FASE DI ELABORAZIONE Viene programmato il nucleo, rischioso, dell'architettura

TRUE



CORRETTO !

## 67. Domanda #67

In UML per responsabilità si intende la specifica di un metodo associato ad una classe Java

FALSE



CORRETTO !

## 68. Domanda #68

In UML per responsabilità si intende la specifica di una variabile di istanza di una classe Java

FALSE



CORRETTO !

## 69. Domanda #69

In UML per responsabilità si intende un contratto o un obbligo di un classificatore

TRUE



CORRETTO !

## 70. Domanda #70

In UML per responsabilità si intende la specifica di una variabile di istanza o di un metodo associato ad una classe Java

FALSE



CORRETTO !

## 71. Domanda #71

Gli SSD Sono espressi attraverso i diagrammi di comunicazione di UML

FALSE



CORRETTO !

## 72. Domanda #72

Gli SSD Mostrano l'ordine degli eventi generati dagli attori esterni al sistema

TRUE



CORRETTO !

## 73. Domanda #73

I contratti costituiscono un input per gli SSD delle operazioni e per la progettazione degli oggetti

FALSE



CORRETTO !

## 74. Domanda #74

Un evento di sistema è un evento interno al sistema

FALSE



CORRETTO !

## 75. Domanda #75

Gli SSD mostrano gli eventi generati dagli attori esterni al sistema

TRUE



CORRETTO !

## 76. Domanda #76

In percentuale sul totale dei requisiti funzionali di un progetto, quanti sono quelli che dovrebbero essere identificati entro la conclusione della fase di ideazione?? 50%

70%



CORRETTO !

## 77. Domanda #77

In percentuale sul totale dei requisiti funzionali di un progetto, quanti sono quelli che dovrebbero essere identificati entro la conclusione della fase di elaborazione?? Quasi

100%



CORRETTO !

## 78. Domanda #78

Il riuso white-box è preferibile al riuso black-box

FALSE



CORRETTO !

## 79. Domanda #79

Il meccanismo di ereditarietà non è preferibile al meccanismo di delega per ottenere il riuso del software

TRUE



CORRETTO !

## 80. Domanda #80

L'ereditarietà si avvale del meccanismo di delega

FALSE



**CORRETTO !**

## 81. Domanda #81

Il meccanismo di composizione non rispetta l'incapsulamento

FALSE



**CORRETTO !**

## 82. Domanda #82

Le operazioni di sistema possono essere identificate mentre si abbozzano gli DSD

FALSE



**CORRETTO !**

## 83. Domanda #83

I contratti descrivono in modo dettagliato i cambiamenti richiesti dall'esecuzione di una operazione di sistema, descrivono come devono essere ottenuti questi risultati

FALSE



**CORRETTO !**

## 84. Domanda #84

I pattern GoF incentivano l'uso dell'ereditarietà come meccanismo di riuso del codice, in particolare attraverso il pattern composite

FALSE



**CORRETTO !**

## 85. Domanda #85

I pattern GoF prediligono l'utilizzo del meccanismo di ereditarietà per ottenere la specializzazione

FALSE



**CORRETTO !**

## 86. Domanda #86

I pattern GoF incentivano i meccanismi di riuso del software attraverso la definizione di gerarchie di classi

FALSE



**CORRETTO !**

## 87. Domanda #87

I pattern GoF prediligono l'utilizzo del meccanismo di ereditarietà per ottenere il polimorfismo

TRUE



**CORRETTO !**

## 88. Domanda #88

I pattern GoF composite fornisce caratteristiche addizionali ad elementi atomici (foglie), mantenendo una interfaccia comune

FALSE



**CORRETTO !**

**89. Domanda #89**

Il pattern GoF composite permette di costruire tassonomie di classi

 FALSE**CORRETTO !****90. Domanda #90**

Il pattern GoF composite utilizza il meccanismo di specializzazione per rendere l'interfaccia delle entità atomiche esattamente come l'interfaccia delle entità composte

 FALSE**CORRETTO !****91. Domanda #91**

Il pattern GoF composite permette di evitare l'esplosione delle sottoclassi per supportare un ampio numero di estensioni e combinazioni di esse

 FALSE**CORRETTO !****92. Domanda #92**

Il pattern GoF composite permette di costruire strutture ricorsive in modo che ad un cliente l'intera struttura sia vista come una singola entità

 TRUE**CORRETTO !****93. Domanda #93**

L'extreme programming promuove lo sviluppo seguito dai test

 FALSE**CORRETTO !****94. Domanda #94**

I test unitari hanno lo scopo di verificare la comunicazione tra specifiche parti del sistema

 FALSE**CORRETTO !****95. Domanda #95**

Preparazione, esecuzione, verifica e rilascio sono le parti del test unitario

 TRUE**CORRETTO !****96. Domanda #96**

Il refactoring prevede di applicare piccole trasformazioni che preservano il comportamento

 TRUE**CORRETTO !****97. Domanda #97**

I test unitari sono utilizzati per dimostrare che il refactoring non abbia causato una regressione

TRUE



CORRETTO !

**98. Domanda #98**

La progettazione object-oriented in UP ha il fine di identificare le responsabilità e assegnarle ad un opportuno oggetto o classe

TRUE



CORRETTO !

**99. Domanda #99**

La coesione è la misura di quanto fortemente un elemento è connesso ad altri elementi, ha conoscenza di altri elementi e dipende da altri elementi

FALSE



CORRETTO !

**100. Domanda #100**

Il pattern GRASP Information Expert e High Cohesion sono pattern valutativi

FALSE



CORRETTO !

**101. Domanda #101**

La creazione di un oggetto è di responsabilità di un oggetto che aggrega, contiene un'istanza dell'oggetto da creare

TRUE



CORRETTO !

**102. Domanda #102**

Una classe con accoppiamento alto fa molte cose non correlate tra loro e svolte troppo lavoro

FALSE



CORRETTO !

**103. Domanda #103**

Il refactoring è una pratica promossa dal metodo iterativo e agile xp

TRUE



CORRETTO !

**104. Domanda #104**

Il refactoring prevede lo sviluppo guidato dai test, ovvero uno sviluppo preceduto dai test

FALSE



CORRETTO !

**105. Domanda #105**

Il refactorin è un metodo strutturato e disciplinato per scrivere o ristrutturare del codice esistente

TRUE



CORRETTO !

**106. Domanda #106**

DURANTE LA FASE DI ELABORAZIONE: I requisiti e le iterazioni sono organizzati in base alla richiesta dell'utente finale

FALSE

**CORRETTO !****107. Domanda #107**

DURANTE LA FASE DI ELABORAZIONE: I requisiti e le iterazioni sono organizzati in base al rischio, copertura (devono coprire tutto il sistema) e criticità

TRUE

**CORRETTO !****108. Domanda #108**

DURANTE LA FASE DI ELABORAZIONE: Si effettua attività di programmazione di qualità-produzione e test

TRUE

**CORRETTO !****109. Domanda #109**

DURANTE LA FASE DI ELABORAZIONE: Vengono realizzati prototipi “usa e getta” per ottenere i rischi maggiori

FALSE

**CORRETTO !****110. Domanda #110**

La responsabilità è un'astrazione di ciò che ha o rappresenta un oggetto o un componente software

TRUE

**CORRETTO !****111. Domanda #111**

Nel responsibility-Drive Developement gli oggetti software sono considerati come dotati di responsabilità

TRUE

**CORRETTO !****112. Domanda #112**

La disciplina dei requisiti è il processo per scoprire cosa deve essere costruito

TRUE

**CORRETTO !****113. Domanda #113**

La disciplina dei requisiti deve orientare lo sviluppo verso il sistema corretto

TRUE

**CORRETTO !****114. Domanda #114**

I contratti usano precondizioni e post-condizioni per descrivere nel dettaglio i cambiamenti agli oggetti in un modello di progetto (software)

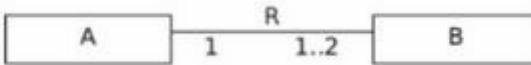
FALSE



CORRETTO !

**115. Domanda #115**

Si consideri la seguente associazione R tra a e B:

se  $A = \{a_1, a_2, a_3\}$  e  $B = \{b_1, b_2\}$ , R può essere  $\{(a_1, b_1), (a_2, b_2)\}$ 

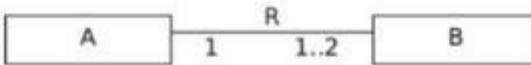
FALSE



CORRETTO !

**116. Domanda #116**

Si consideri la seguente associazione R tra a e B:

se  $A = \{a_1, a_2\}$  e  $B = \{b_1, b_2, b_3\}$ , R può essere  $\{(a_1, b_1), (a_2, b_2), (a_1, b_2), (a_2, b_3)\}$ 

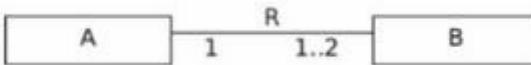
FALSE



CORRETTO !

**117. Domanda #117**

Si consideri la seguente associazione R tra a e B:

se  $A = \{a_1, a_2\}$  e  $B = \{b_1, b_2, b_3\}$ , R può essere  $\{(a_1, b_1), (a_1, b_2), (a_2, b_3)\}$ 

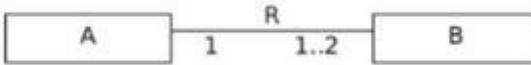
TRUE



CORRETTO !

**118. Domanda #118**

Si consideri la seguente associazione R tra a e B:

se  $A = \{a_1, a_2\}$  e  $B = \{b_1, b_2\}$ , R può essere  $\{(a_1, b_1), (a_1, b_2)\}$ 

FALSE

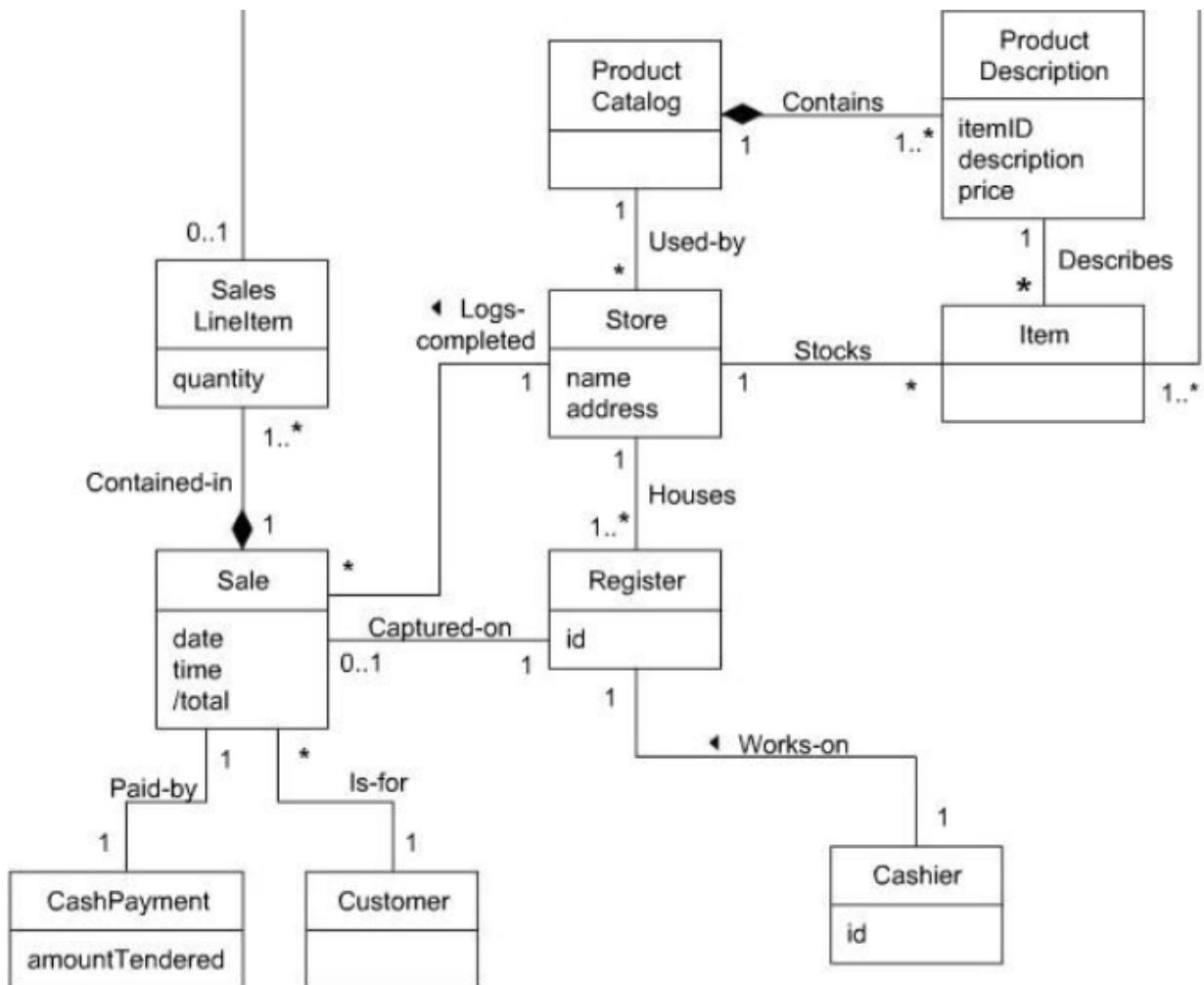


CORRETTO !

**119. Domanda #119**

Si consideri l'associazione "Contained-in" nel seguente Modello di Dominio:

Records-sale-of



Una istanza di Sale deve essere distrutta prima di un'istanza di SalesLineItem

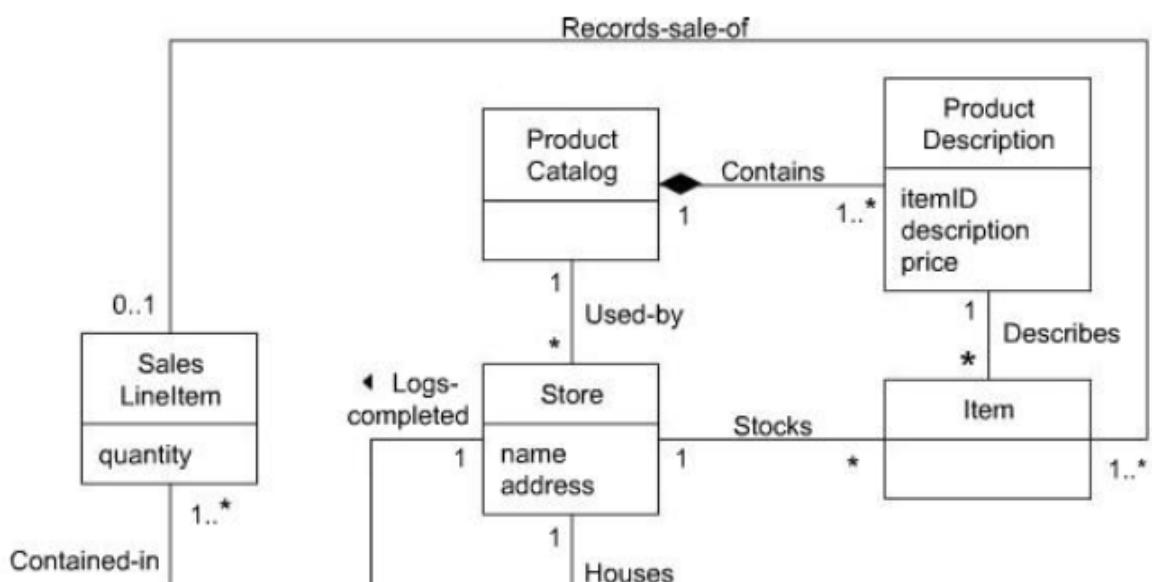
FALSE  CORRETTO !

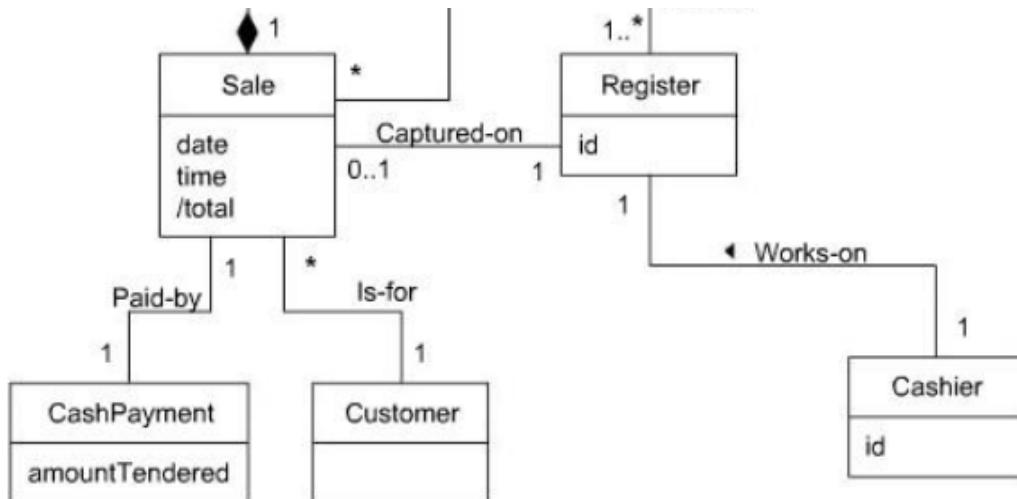
#### Motivo:

l'istanza Sale è l'intero ( il contenitore) e la parte è SalesLineItem. Dato che siamo in una composizione, la parte (SalesLineItem) non può esistere senza l'intero e quindi non posso distruggere l'istanza intero (Sale) prima di un'istanza parte.

## 120. Domanda #120

Si consideri l'associazione “Contained-in” nel seguente Modello di Dominio:





Una istanza di SalesLineItem deve essere distrutta prima di un'istanza di Sale.

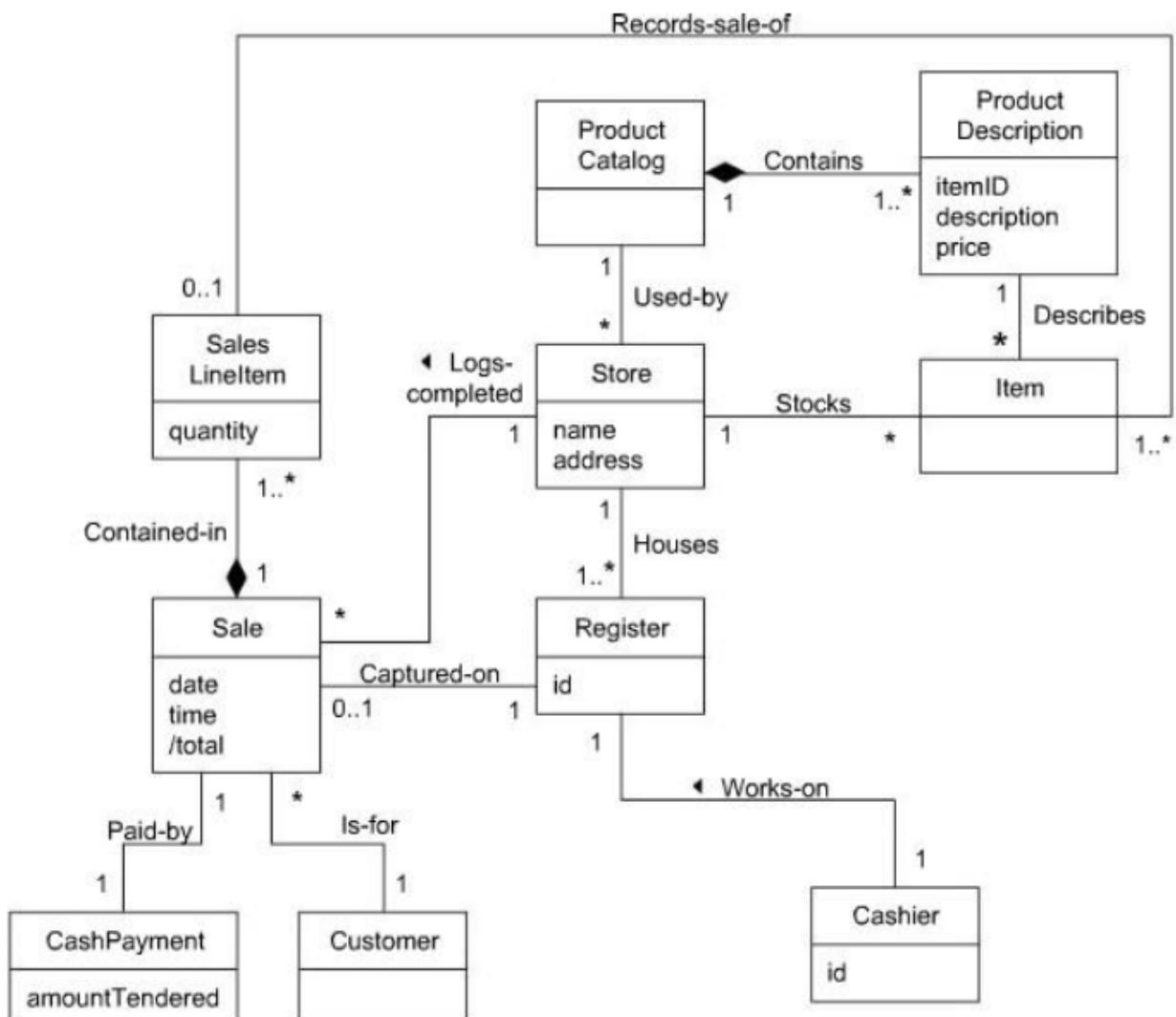
TRUE CORRETTO !

### Motivo:

è il contrario di quanto detto prima (domanda 122)

## 121. Domanda #121

Si consideri l'associazione “Contained-in” nel seguente Modello di Dominio:



Le istanze di SalesLineItem appartengono ad una sola istanza di Sale alla volta.

TRUE



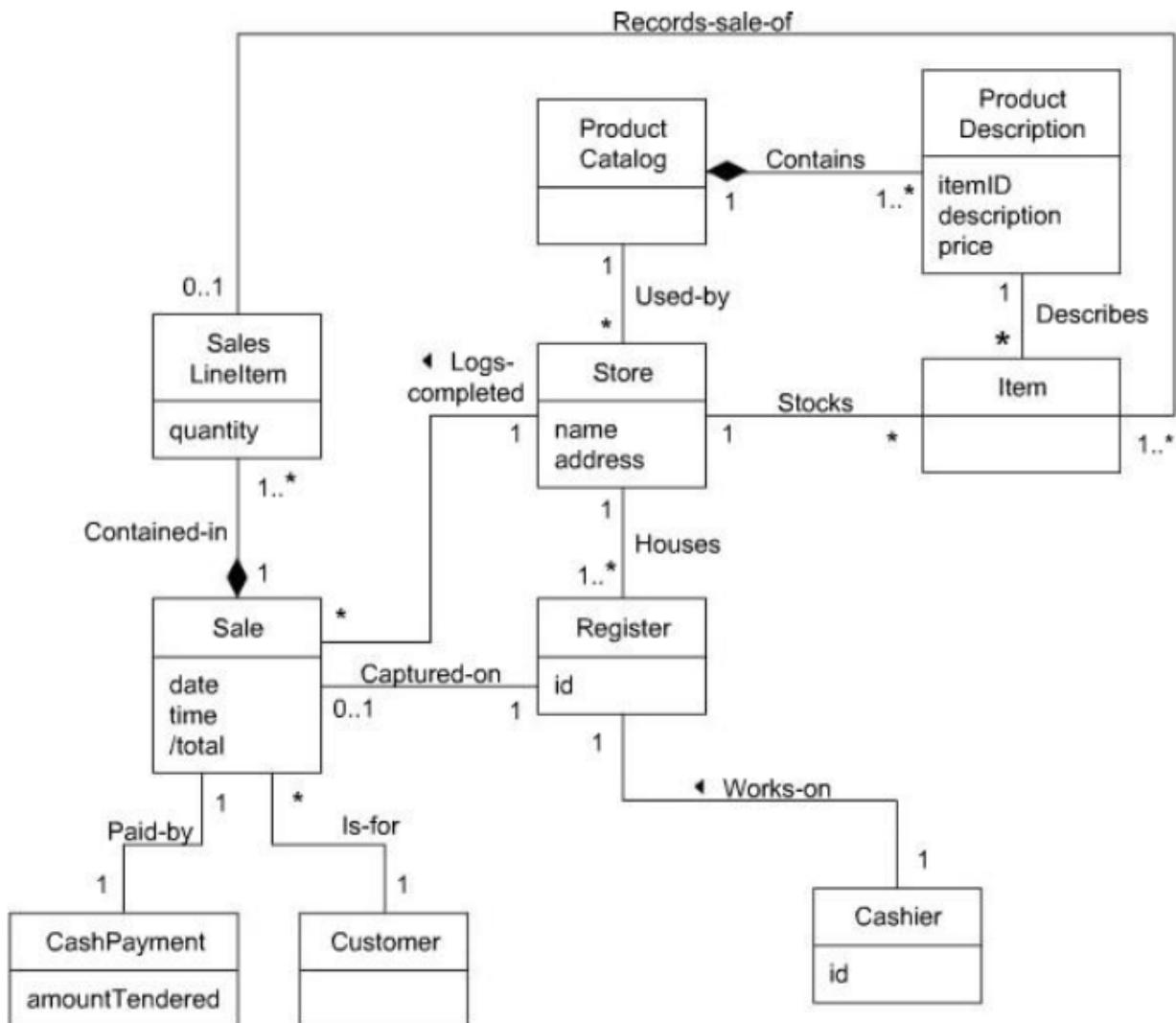
CORRETTO !

**Motivo:**

data dalla cardinalità dell'associazione "contained-in"

**122. Domanda #122**

Si consideri l'associazione "Contained-in" nel seguente Modello di Dominio:



Le istanze di Sale appartengono ad una sola istanza di SalesLineItem alla volta.

FALSE



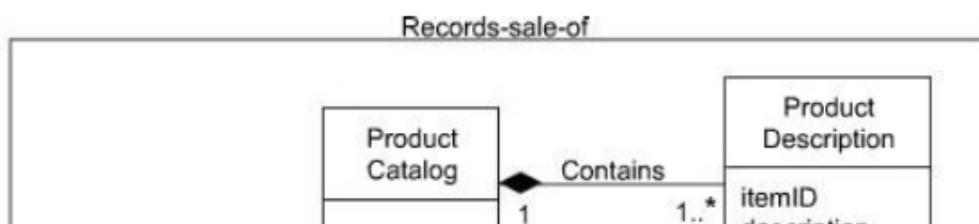
CORRETTO !

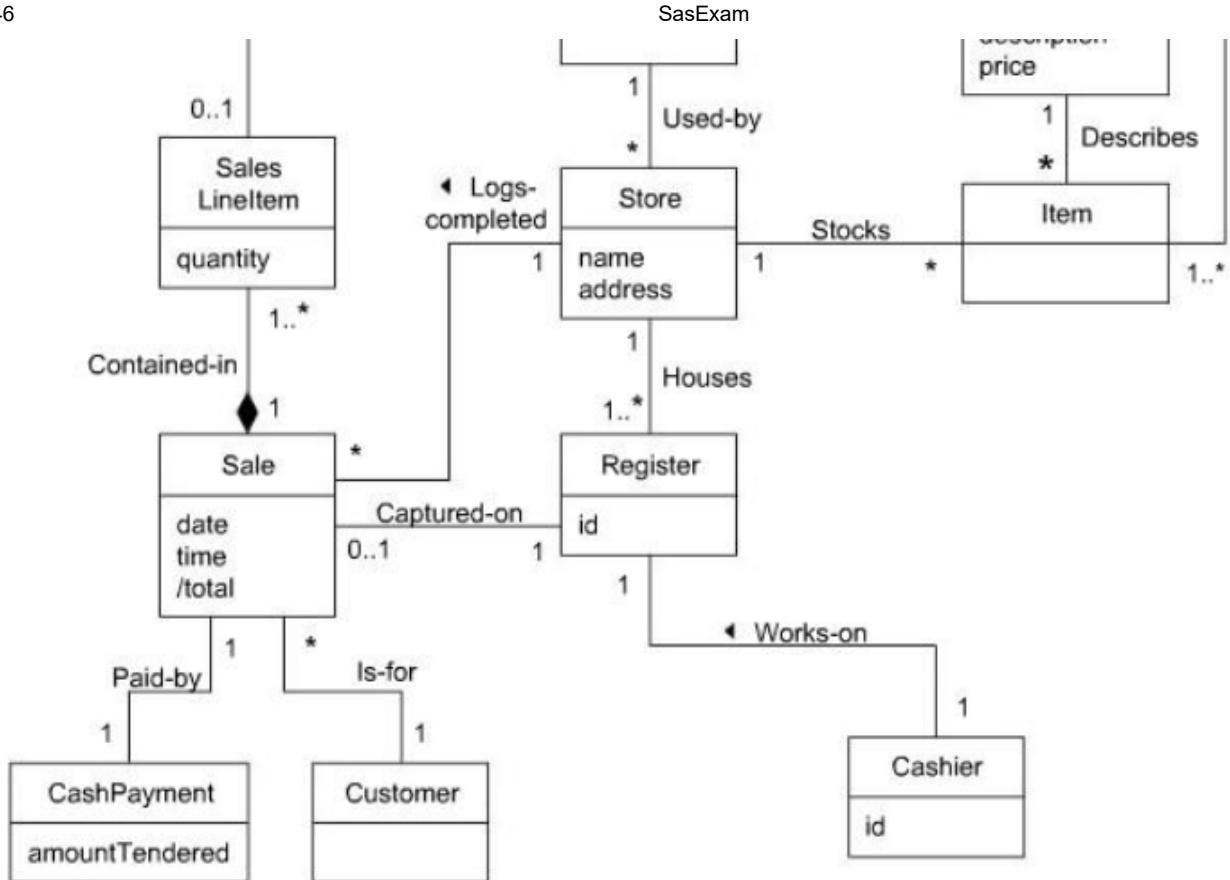
**Motivo:**

perché la cardinalità dell'associazione "contained-in" è 1..\*

**123. Domanda #123**

Si consideri l'associazione "Contained-in" nel seguente Modello di Dominio:





Una istanza di SalesLineItem puo' essere creata dopo un'istanza di Sale.

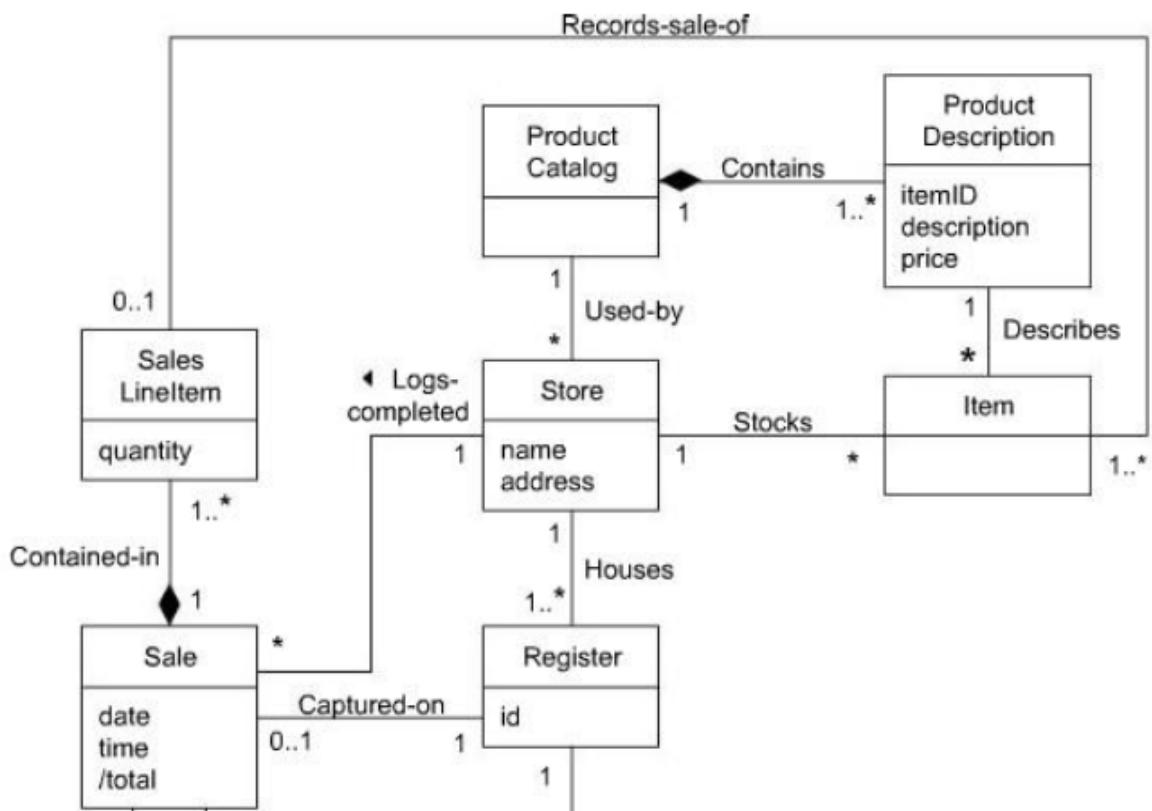
TRUE  CORRETTO !

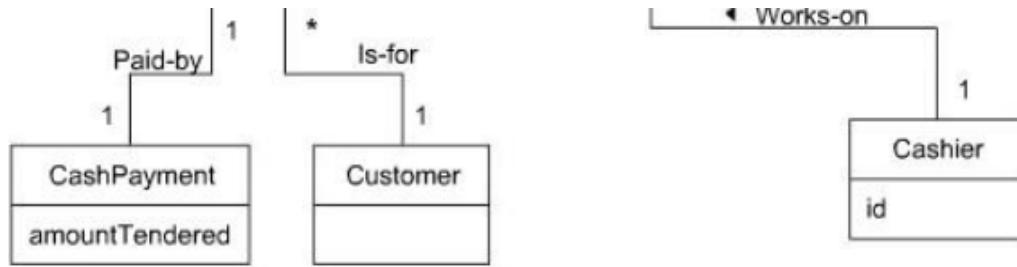
#### Motivo:

per quanto detto in precedenza sulla composizione (domanda 122)

## 124. Domanda #124

Si consideri l'associazione "Contained-in" nel seguente Modello di Dominio:





Una istanza di Sale pu`o essere creata dopo un'istanza di SalesLineItem.

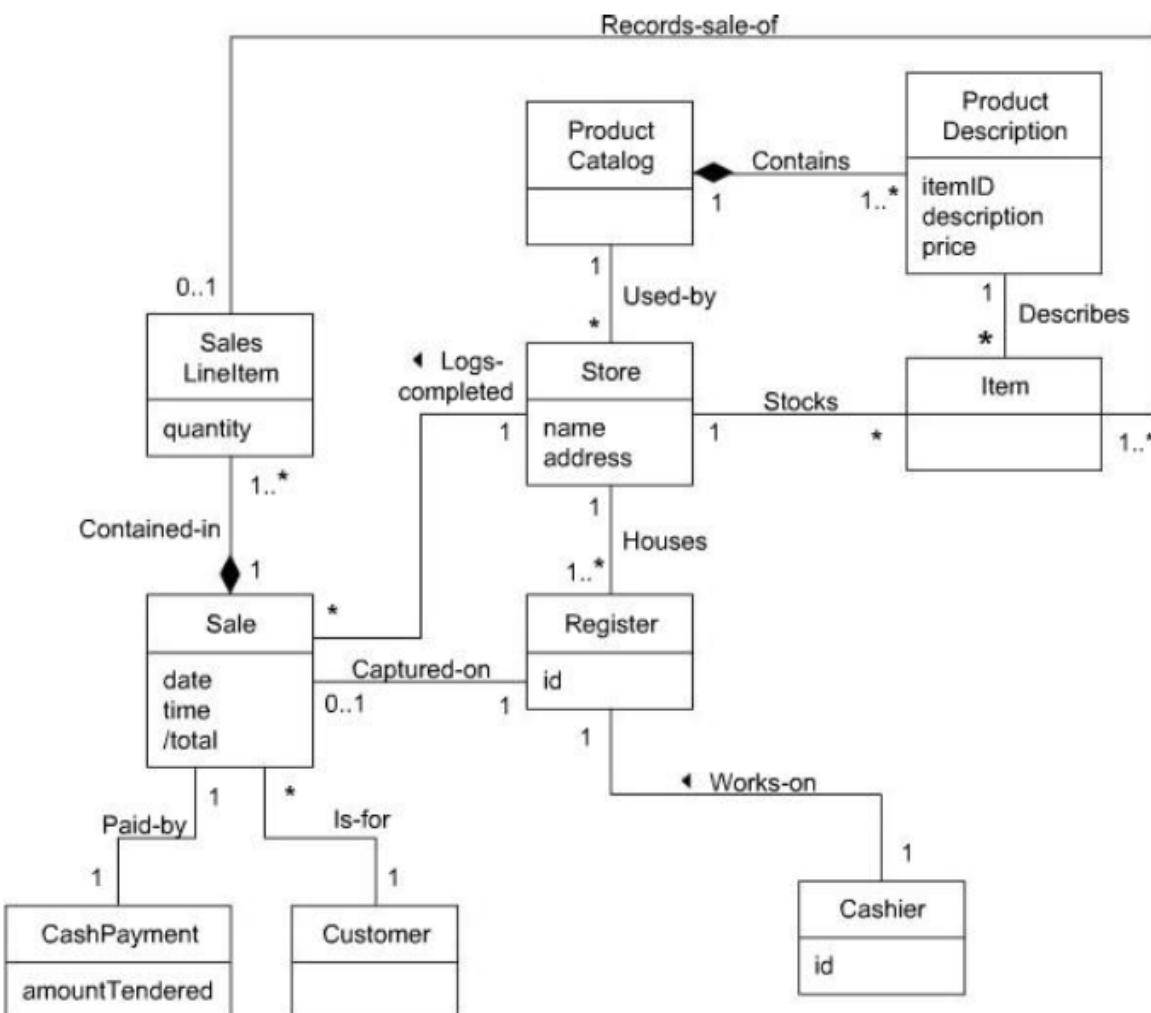
FALSE CORRETTO !

**Motivo:**

per quanto detto in precedenza sulla composizione (domanda 122)

## 125. Domanda #125

Si consideri il seguente Modello di Dominio: Si supponga Product Catalog = {c1, c2} e Product Description = {d1, d2, d3, d4}.



Contains pu`o essere {(c1, d1), (c1, d2), (c2, d2), (c2, d3), (c2, d4)}.

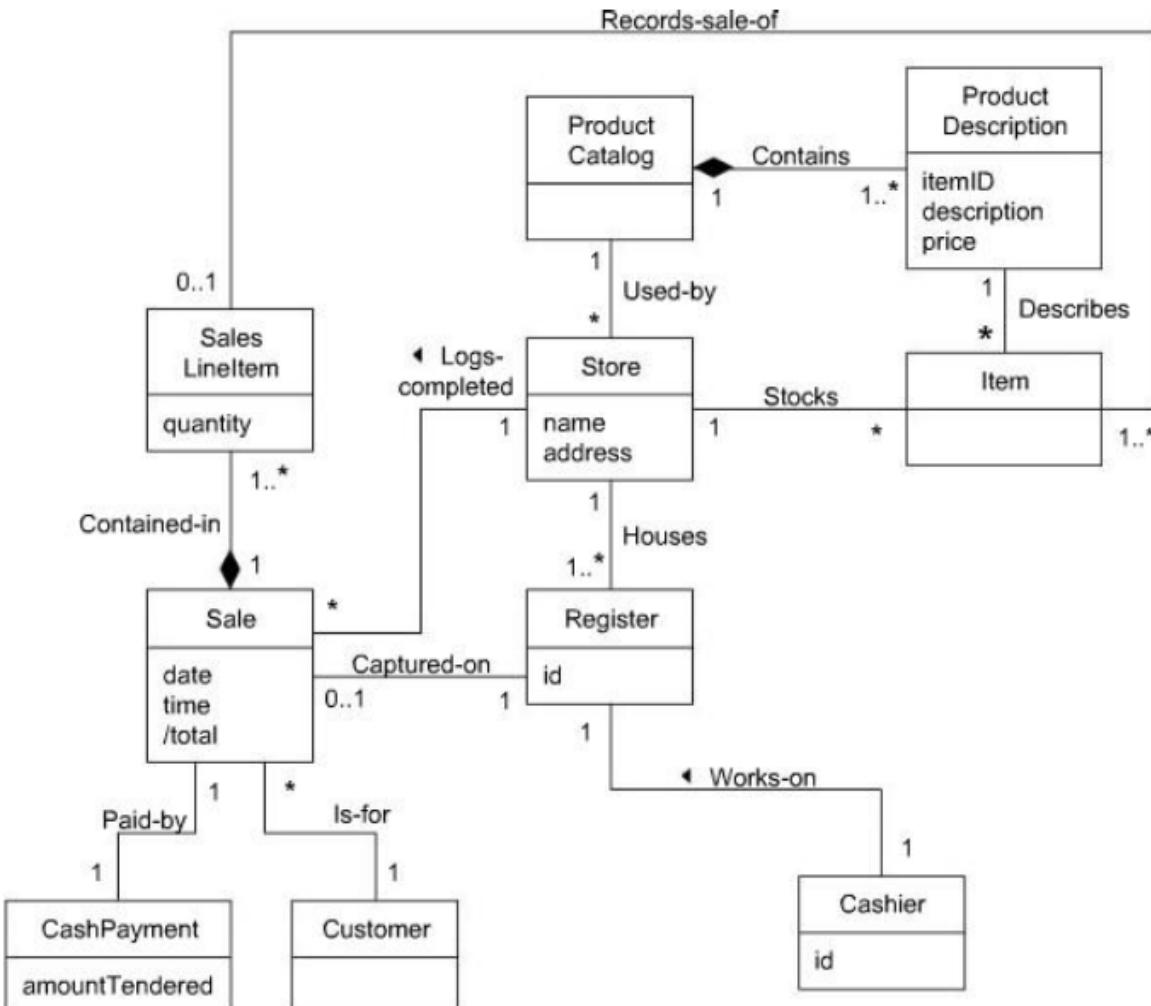
FALSE CORRETTO !

**Motivo:**

d2 è associato a 2 elementi di c (c1 e c2) e questo non va bene perché dev'essere associato solo ad un elemento di c per la cardinalità dell'associazione (1 -- 1..\*)

## 126. Domanda #126

Si consideri il seguente Modello di Dominio: Si supponga Product Catalog = {c1, c2} e Product Description = {d1, d2, d3, d4}.



Contains pu`o essere  $\{(c1, d1), (c1, d2), (c1, d3), (c2, d4)\}$ .

TRUE



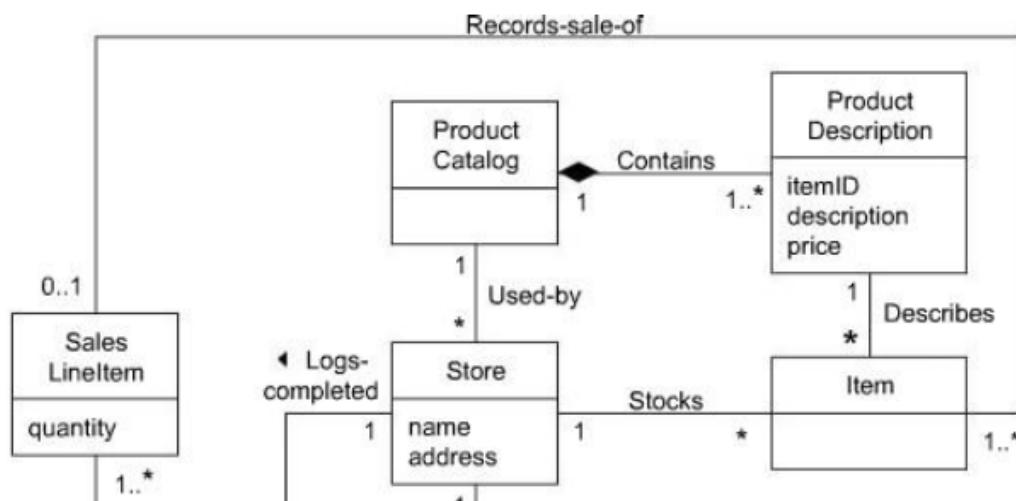
**CORRETTO !**

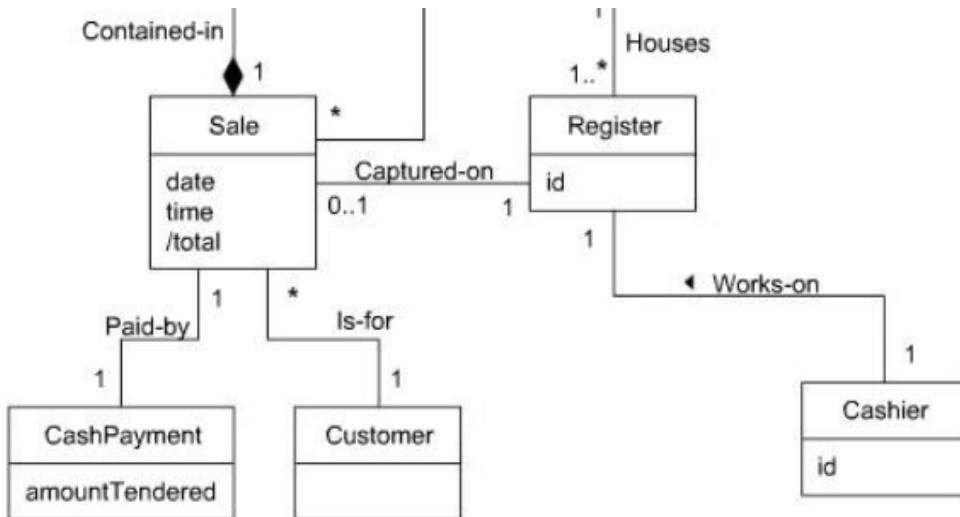
### Motivo:

vengono rispettate le cardinalità

## 127. Domanda #127

Si consideri il seguente Modello di Dominio: Si supponga Product Catalog = {c1, c2} e Product Description = {d1, d2, d3, d4}.





Contains pu`o essere  $\{(c1, d1), (c1, d2), (c2, d3)\}$ .

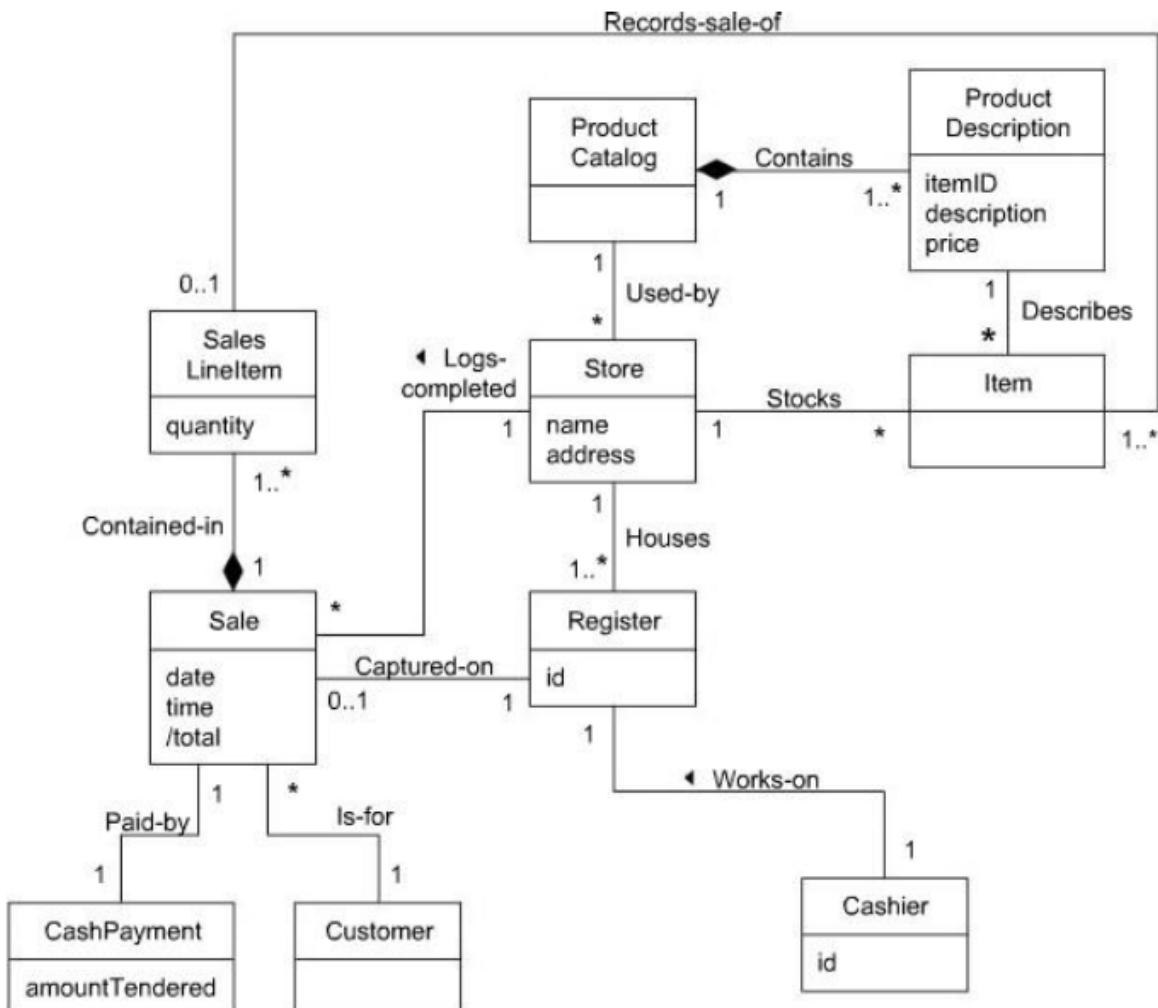
FALSE CORRETTO !

### Motivo:

perché d3 non è associato a nessun elemento di c

## 128. Domanda #128

Si consideri il seguente Modello di Dominio:



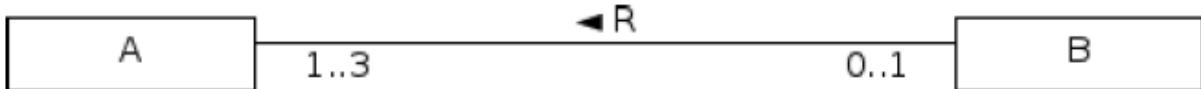
Contains pu`o essere  $\{(c1, d1), (c1, d2), (c1, d3), (c1, d4)\}$ .

FALSE CORRETTO !

**Motivo:** c2 non è associato a nessun elemento di d

## 129. Domanda #129

Si consideri il diagramma seguente e Si supponga A = {a1, a2, a3, a4} e B = {b1, b2}.



R pu`o essere {(b1, a1), (b1, a2), (b2, a1)}.

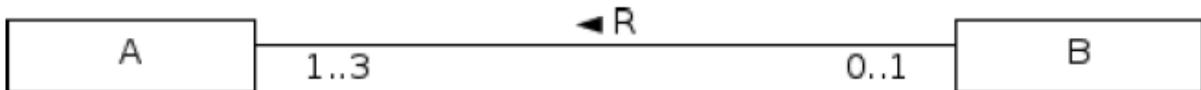
FALSE  CORRETTO !

**Motivo:**

perché a1 è associato a 2 elementi di b, quando invece la cardinalità è (0..1).

## 130. Domanda #130

Si consideri il diagramma seguente e Si supponga A = {a1, a2, a3, a4} e B = {b1, b2}.



R pu`o essere {(b1, a1), (b1, a2), (b1, a3)}.

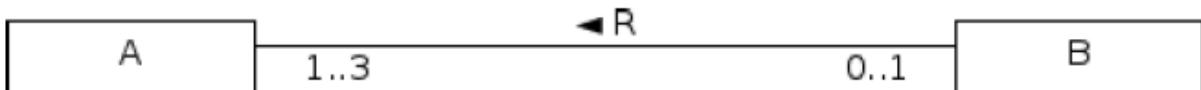
FALSE  CORRETTO !

**Motivo:**

perché b2 non è associato a nessun elemento di a, quando invece la cardinalità è (1..3)

## 131. Domanda #131

Si consideri il diagramma seguente e Si supponga A = {a1, a2, a3, a4} e B = {b1, b2}.



R pu`o essere {(b1, a1), (b1, a2), (b1, a3), (b1, a4), (b2, a3)}.

FALSE  CORRETTO !

**Motivo:**

perché a3 è associato a 2 elementi di b, quando invece la cardinalità è (0..1)

## 132. Domanda #132

Si consideri il diagramma seguente e Si supponga A = {a1, a2, a3, a4} e B = {b1, b2}.



1..3

R pu`o essere  $\{(b1, a1), (b1, a2), (b2, a3), (b2, a4)\}$ .

TRUE



CORRETTO !

**Motivo:**

tutto ok, le cardinalità sono rispettate