

```

`a
à
è
oggetto
à
classe
è
E
Attributi
Metodi
oggetto
è
new
O
è
private
public
protected
Analisi
del
codice
della
classe
Per-
sona:
Persona
u
Attributi
pri-
vati
(righe
44-
47):
nome
cognome
età
private
in-
cap-
su-
la-
mento
Costruttori
(righe
49-
61):
over-
load-
ing
this
è
this
nome
nome
Metodi
get-
ter
(righe
63-
74):
è
Persona
private
persona
Metodi
set-
ter
(righe
76-
81):
persona.setEta()
à
à
persona
è
è
val-
i-
dazione
dei
dati
Metodi
di
utilità
(righe
84-
91):
persona.stampaInfo()
persona
persona.isMaggiorenne()
Analisi
del
codice
di
Test-
Per-
sona:
Persona
Creazione
di
oggetti
(righe
101-
103):
new

```

```

p3.eta
30
è
p3.eta
è
p3.setEta()
Analisi
del
codice
della
classe
Auto:
Auto
u
Inizializzazione
della
ve-
locità
(riga
134):
auto.velocitaAttuale
è
auto
è
Metodo
ac-
cel-
era
(righe
137-
143):
auto.accelera()
auto
a
if
(incremento
>
0)
è
+=
è
velocitaAttuale
velocitaAttuale
incremento
a
a
Metodo
frena
(righe
145-
151):
auto.frena()
è
Math.max()
java.lang.Math
Math.max(0,
velocitaAttuale
-decremento)
a
Math.max()
è
a
Metodo
get-
De-
scrizione
(righe
153-
155):
auto.getDescrizione()
u
auto
E
è
Auto
com-
po-
sizione
ag-
gregazione

```

Restricted shell escape. PlantUML cannot be called. Start pdf_latex/lualatex with -shell-escape. @startuml skinparam classAttributeIconSize 0 skinparam classFontSize 12 skinparam ArrowColor 3366cc skinparam ClassBackgroundColor E6F0FF skinparam ClassBorderColor 3366cc

```

' === CLASSI ===
class Auto - String marca - Motore motore - + Auto()
+ accendi()
class Motore - int cilindrata - int cavalli - + avvia() + spegni()
class Scuola - String nome - Studente[] studenti - + aggiungiStudente() +
rimuoviStudente()
class Studente - String matricola - String nome - + studia() + sostieneE-
same()
' === RELAZIONI ===
Auto *- Motore : composizione(parte-tutto)
Scuola o- Studente : aggregazione(contenimento)
' === NOTE ESPLICATIVE ===
note bottom of Auto Composizione
(diamante pieno): il Motore non può esistere senza Auto. Se l'Auto viene
distrutta, anche il Motore cessa di esistere. end note
note bottom of Scuola Aggregazione (diamante vuoto): gli Studenti esistono
indipendentemente dalla Scuola. Uno Studente può cambiare scuola o esistere
senza scuola. end note
@enduml

```