UNIVERSITÀ DI PISA

Corso di Laurea in Ingegneria Informatica

c/o Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica, Telecomunicazioni

Esami di Ingegneria del Software
e di Ingegneria dei Sistemi Software

Appello dell'8 febbraio 2005

Nome e cognome:
Matricola:
Esame: **Ing. del Software** ☐ **Ing. dei Sistemi Software** ☐

Il punteggio in trentesimi associato a ciascuna domanda è indicato fra parentesi. Ove si trovino coppie di domande con lo stesso numero, rispondere solo a quella corrispondente all'esame che si vuole sostenere: la domanda per l'esame di Ingegneria del Software (vecchio ordinamento) è identificata dalla sigla [**V**], quella per l'esame di Ingegneria dei Sistemi Software (nuovo ordinamento) dalla sigla [**N**].

Scegliere una risposta per ciascuna domanda:

1 Quale di queste stringhe viene riconosciuta dall'ER `'[1-9].[0-9]+[eE](+-)?[0-9]+'`? (1)
3.14e1 ☒
31.4e-1 ☐
3.14+1 ☐

2 La formula $A \Rightarrow \neg A$ è (1)
vera. ☐
soddisfacibile. ☒
valida. ☐

3 In un sistema formale corretto (1)
tutte le formule dimostrabili sono vere. ☒
tutte le formule vere sono dimostrabili. ☐
tutti gli assiomi sono dimostrabili. ☐

4 Nelle Reti di Petri l'abilitazione di una transizione dipende (1)
dalla marcatura dei place di ingresso. ☒
dalla marcatura dei place di uscita. ☐
dalla marcatura sia dei place di ingresso che di uscita. ☐

5 Il test funzionale è basato (1)
sulle specifiche. ☒
sul documento di progetto. ☐
sul codice sorgente. ☐

Rispondere alle domande, usando solo lo spazio disponibile:

6 Che cosa significa la formula $M[t\rangle \wedge M[u\rangle \wedge \forall_{p \in {}^\bullet t \cap {}^\bullet u}(M(p) \geq W(\langle p,t\rangle) + W(\langle p,u\rangle))$? (3)

Le transizioni t ed u sono abilitate nella marcatura M e per ogni place p di ingresso a t ed u si ha che la sua marcatura è non non minore della somma dei pesi degli archi che lo uniscono a t ed u. Le due transizioni sono quindi in *concorrenza effettiva.*

7 Quali tipi di eventi sono previsti in UML? (3)

Chiamate di operazione, cambiamenti, segnali, eventi temporali (relativi o assoluti).

8 Dimostrare che le formule $A \wedge \neg B$ e $\neg(A \Rightarrow B)$ sono equivalenti. (3)

A	B	$A \wedge \neg B$	$\neg(A \Rightarrow B)$
T	**T**	F	$\neg(T \Rightarrow T) = F$
T	**F**	T	$\neg(T \Rightarrow F) = T$
F	**T**	F	$\neg(F \Rightarrow T) = F$
F	**F**	F	$\neg(F \Rightarrow F) = F$

9 Con quali diagrammi UML si descrivono le interazioni fra oggetti? (3)

Diagrammi di sequenza e di collaborazione.

10 [V] A cosa servono i classificatori ≪interface≫? (3)

Un ≪interface≫ permette di rappresentare un'interfaccia indipendentemente dalle entità (classi, package, componenti) che la implementano.

10 [N] Che cosa sono le sequenze (`sequence<>`) nel linguaggio IDL? (3)

Array unidimensionali con un numero variabile di elementi.

11 Con riferimento alla Fig. 1, rispondere alle domande. (5)

	V	F
il **Produttore** inizia l'interazione	☐	☒
il **Produttore** fa terminare l'interazione	☐	☒
il **Produttore** invia il segnale consuma	☒	☐
il **Consumatore** entra in **Elaborazione** prima del **Produttore**	☐	☒
il **Consumatore** entra in **Attesa** quando riceve produci	☐	☒

12 Con riferimento al codice di Fig. 2, disegnare un diagramma di classi che lo descriva. (5)

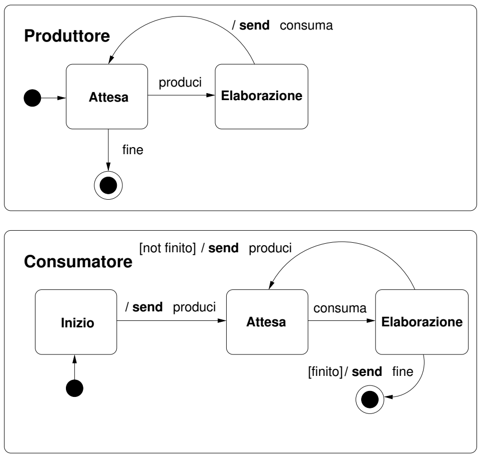

Figura 1: Domanda 11.

```
class ServantBase {
public:
    virtual void invoke() = 0;
};
class Dispatcher {
    ServantBase* srvt_;
public:
    void set_servant(ServantBase* s);
    void dispatch();
};
class POA_Greeting : public ServantBase {
public:
    virtual void say_hello() = 0;
    virtual void invoke();
};
class Greeting_impl : public POA_Greeting {
    virtual void say_hello();
};
```

```
void Dispatcher::
set_servant(ServantBase* s)
{
    srvt_ = s;
}
void Dispatcher::
dispatch()
{
    srvt_->invoke();
}
void POA_Greeting::
invoke()
{
    this->say_hello();
}
void Greeting_impl::
say_hello()
{
    cout << "Hello!" << endl;
}
```

Figura 2: Domanda 12.

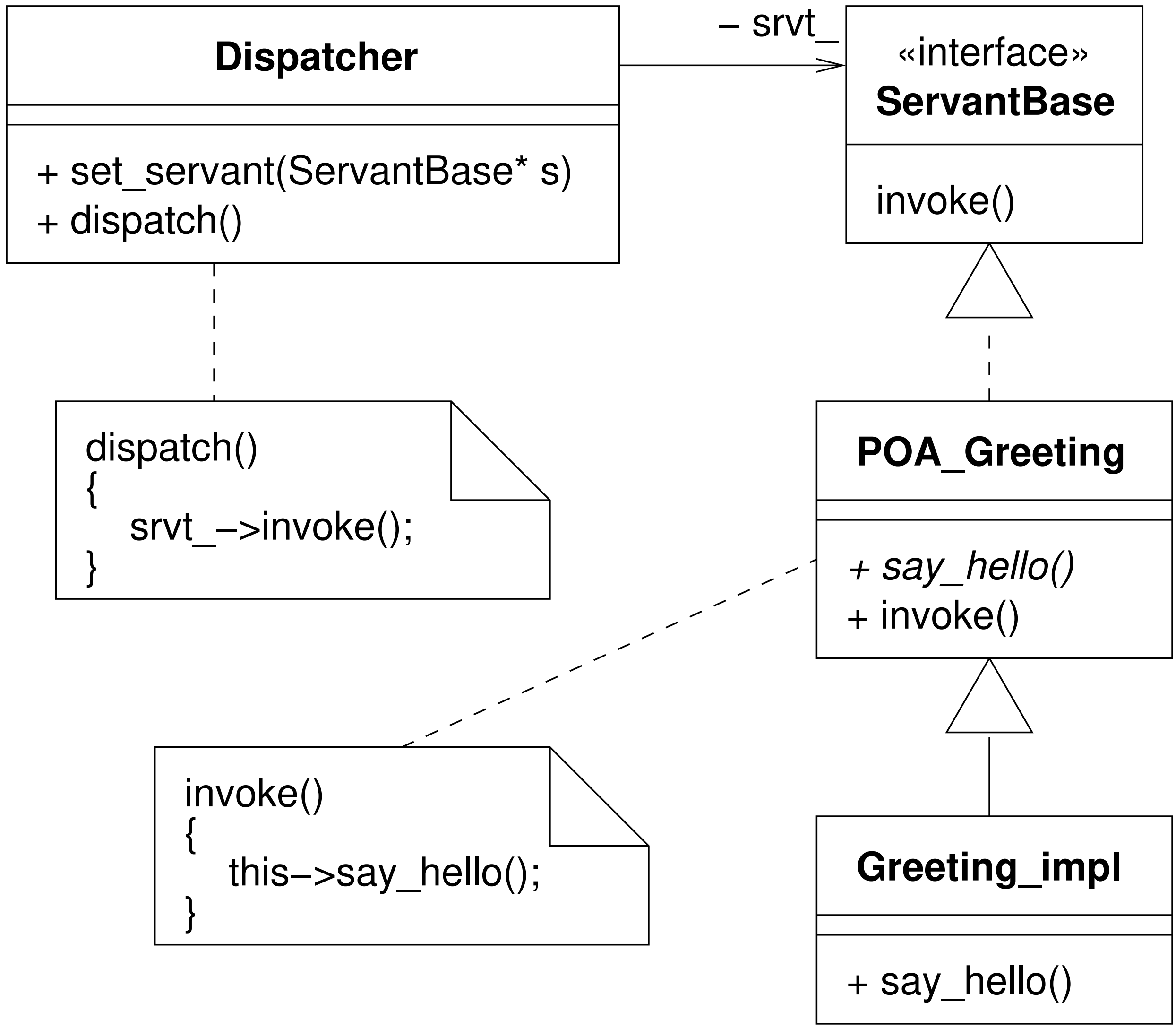

Figura 3: Domanda 12, soluzione.