

[Home](#) / [I miei corsi](#) / [SAS-EE-2122](#) / [Esonero del 15 giugno 2022](#) / [Esonero del 15 giugno 2022](#)

Iniziato mercoledì, 15 giugno 2022, 10:14

Stato Completato

Terminato mercoledì, 15 giugno 2022, 10:48

Tempo impiegato 34 min. 14 secondi

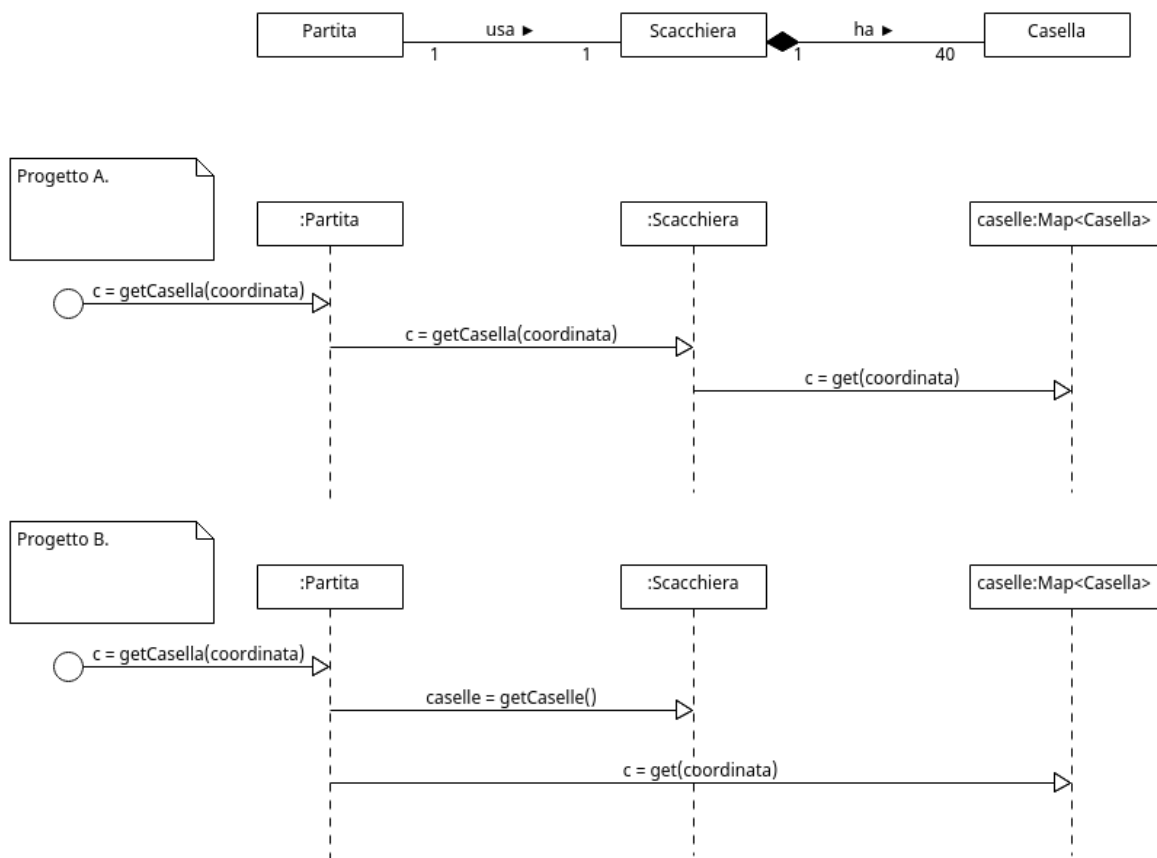
Valutazione 8,66 su un massimo di 14,00 (61,88%)

Domanda **1**

Completo

Punteggio ottenuto 1,25 su 2,50

Si consideri il modello di dominio (in alto) e i due progetti di interazione, denominati progetto A e progetto B. Si dica quali delle seguenti affermazioni sono vere e quali false.



Il progetto A presenta 2 sole dipendenze/accoppiamenti: Partita -> Scacchiera e Scacchiera -> Casella.

Vero

Il progetto A prevede la delega di getCasella da Partita a Scacchiera.

Vero

Il progetto B presenta 2 sole dipendenze/accoppiamenti: Partita -> Scacchiera e Partita -> Casella.

Vero

Il progetto migliore secondo il principio di low coupling è il progetto A.

Falso

Domanda **2**

Completo

Punteggio ottenuto 2,25 su 2,25

Si consideri il seguente codice:

```
public interface CompressionAlgorithm {
    public void compressFiles(ArrayList<File> files);
}

public class ZipCompressionAlgorithm implements CompressionAlgorithm {
    public void compressFiles(ArrayList<File> files) {
        //using ZIP algorithm
    }
}

public class RarCompressionAlgorithm implements CompressionAlgorithm {
    public void compressFiles(ArrayList<File> files) {
        //using RAR algorithm
    }
}

public class CompressionContext {
    private CompressionAlgorithm algorithm;
    //this can be set at runtime by the application preferences
    public void setCompressionAlgorithm(CompressionAlgorithm algorithm) {
        this.algorithm = algorithm;
    }
    public void createArchive(ArrayList<File> files) {
        algorithm.compressFiles(files);
    }
}

public class Client {
    public static void main(String[] args) {
        CompressionContext ctx = new CompressionContext();
        //we could assume context is already set by preferences
        ctx.setCompressionAlgorithm(new ZipCompressionAlgorithm());
        //get a list of files...
        ctx.createArchive(fileList);
    }
}
```

Questo utilizza un insieme di classi che adottano un noto pattern GoF. Si dica di quale pattern si tratta.

Scegli un'alternativa:

- ☐ a. Abstract Factory
- ☒ b. Strategy
- ☐ c. Adapter
- ☐ d. Composite
- ☐ e. Visitor
- ☐ f. Singleton
- ☐ g. Decorator
- ☐ h. State
- ☐ i. Observer

Domanda **3**

Completo

Punteggio ottenuto 1,31 su 1,75

Dire quali delle seguenti affermazioni sono vere e quali false.

Il processo sviluppo incrementale è migliore rispetto al processo a cascata sotto ogni punto di vista e contesto di applicazione.

Lo sviluppo incrementale si basa sull'idea di sviluppare un'implementazione iniziale, esporla agli utenti e perfezionarla attraverso molte versioni.

Il processo a cascata è basata su uno svolgimento sequeziale delle diverse attività dello sviluppo software.

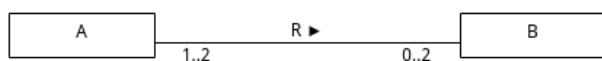
Il processo a cascata è migliore rispetto allo sviluppo incrementale sotto ogni punto di vista e contesto di applicazione.

Domanda **4**

Completo

Punteggio ottenuto 1,20 su 2,00

Si consideri la seguente associazione R tra A e B:



dire quali delle seguenti affermazioni sono vere e quali false.

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

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

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

Le istanze di B sono visibili ad A ma non viceversa.

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

Domanda **5**

Completo

Punteggio ottenuto 0,90 su 1,50

Dire quali delle seguenti affermazioni sono vere e quali false.

Durante l'elaborazione i documenti vengono abbozzati e definiti in maniera "leggera".

La pianificazione delle iterazioni è basata sul rischio, inizialmente vengono scelti i casi d'uso meno rischiosi.

L'ideazione è la prima iterazione dell'elaborazione.

La maggior parte dei requisiti è individuata durante la fase di ideazione.

Il modello dei casi d'uso viene iniziato durante l'elaborazione.

Domanda **6**

Completo

Punteggio ottenuto 0,00 su 2,25

Si consideri il seguente codice:

```
public interface Behavior {

    void onEnter();
    void observe();
}

public class PeacefulBehavior implements Behavior {

    private final Mammoth mammoth;

    public PeacefulBehavior(Mammoth mammoth) {
        this.mammoth = mammoth;
    }
    public void observe() {
        LOGGER.info("{} is calm and peaceful.", mammoth);
    }
    public void onEnter() {
        LOGGER.info("{} calms down.", mammoth);
    }
}

public class AngryBehavior implements Behavior {

    private final Mammoth mammoth;

    public AngryBehavior(Mammoth mammoth) {
        this.mammoth = mammoth;
    }
    public void observe() {
        LOGGER.info("{} is furious!", mammoth);
    }
    public void onEnter() {
        LOGGER.info("{} gets angry!", mammoth);
    }
}

public class Mammoth {

    private Behavior behavior;

    public Mammoth() {
        behavior = new PeacefulBehavior(this);
    }
    public void timePasses() {
        if (behavior.getClass().equals(PeacefulBehavior.class)) {
            changeBehaviorTo(new AngryBehavior(this));
        } else {
            changeBehaviorTo(new PeacefulBehavior(this));
        }
    }
    private void changeBehaviorTo(Behavior newBehavior) {
        this.behavior = newBehavior;
        this.behavior.onEnter();
    }
    public String toString() {
        return "The mammoth";
    }
}
```

```
public void observe() {  
    this.behavior.observe();  
}  
}
```

Questo utilizza un insieme di classi che adottano un noto pattern GoF. Si dica di quale pattern si tratta.

Scegli un'alternativa:

- ☐ a. Abstract Factory
- ☐ b. Adapter
- ☐ c. Strategy
- ☐ d. Singleton
- ☐ e. Composite
- ☒ f. Visitor
- ☐ g. Decorator
- ☐ h. Observer
- ☐ i. State

Domanda **7**

Completo

Punteggio ottenuto 1,75 su 1,75

Dire quali delle seguenti affermazioni sono vere e quali false.

Sia il Controller GRASP che il Controller MVC si occupano di gestire le richieste provenienti dall'utente.

Vero

In generale il Controller GRASP delega le richieste di lavoro dall'utente al Controller MVC della UI.

Falso

Il controller MVC fa parte della UI e gestisce l'interazione con l'utente; la sua implementazione dipende in larga misura dalla tecnologia UI e dalla piattaforma che viene utilizzata.

Vero

[◀ Annunci](#)

Vai a...

[Esonero del 15 giugno 2022 \(copia\) ►](#)