

## Prova d'Esame

Cognome e Nome \_\_\_\_\_ Matricola \_\_\_\_\_

Quesito svolto base ? intermedio ? avanzato ?

Tempo a Disposizione: 4 ore e mezza

### Aste

#### Livello Base

E' necessario scrivere un'applicazione Web che consente di partecipare ad aste in linea. Le aste riguardano oggetti di vario tipo (es: macchine fotografiche, cellulari, automobili, ciclomotori). L'applicazione deve consentire di eseguire il seguente caso d'uso.

#### “Utente effettua rilancio”

##### *Scenario Principale*

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per specificare il codice dell'oggetto per il quale si intende effettuare il rilancio; i codici sono valori interi (1, 2, 3 ecc.)
- Il sistema verifica che nella base di dati esista un oggetto all'asta con il codice specificato. Gli oggetti all'asta sono memorizzati in una tabella **Oggetti** della base di dati che contiene il codice dell'oggetto, la descrizione, l'attuale offerta migliore, il nome e il cognome di chi ha fatto l'offerta.
- Se l'oggetto esiste, l'utente accede ad uno schermo (**offerta.jsp**) nel quale sono riassunti i dati relativi all'oggetto; lo schermo contiene inoltre una maschera attraverso la quale l'utente fornisce (a) il suo nome; (b) il suo cognome; (c) l'importo dell'offerta
- Se l'importo dell'offerta è superiore alla migliore offerta giunta fino a quel punto per l'oggetto, il sistema aggiorna i dati registrando la nuova offerta per l'oggetto con il nome e il cognome dell'utente
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i risultati dell'operazione (dati dell'oggetto e offerta effettuata); lo schermo contiene un collegamento allo schermo iniziale nel caso l'utente voglia eseguire una ulteriore offerta

##### *Scenario alternativo: L'oggetto non esiste*

- L'utente accede ad uno schermo (**errore.jsp**) in cui viene visualizzato un messaggio di errore ed un collegamento allo schermo principale per effettuare una nuova operazione

Scrivere l'applicazione Web che effettua le operazioni elencate sopra secondo le seguenti specifiche:

- L'applicazione deve utilizzare pagine JSP e un'architettura di tipo **Modello 1**.
- La grafica deve essere organizzata utilizzando un foglio di stile **CSS**.
- Non è necessario provvedere alla convalida dei dati sottomessi dall'utente.
- Non è necessario provvedere alla convalida del codice HTML prodotto dalle pagine JSP.

## Livello Intermedio

E' necessario scrivere un'applicazione Web che consente di partecipare ad aste in linea. Le aste riguardano oggetti di vario tipo (es: macchine fotografiche, cellulari, automobili, ciclomotori). L'applicazione deve consentire di eseguire i seguenti caso d'uso.

### “Utente effettua login”

#### *Scenario Principale*

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per fornire nome utente e password
- Il sistema autentica l'utente accedendo ad una tabella **Utenti** della base di dati, che contiene il nome utente, la password, il nome e cognome dell'utente (**NOTA**: è necessario prevedere tra gli altri un utente “Mario Rossi” con nome utente **utente** e password **utente**)
- Se l'autenticazione ha successo, l'utente accede allo schermo principale del caso d'uso successivo (**offerta.jsp**)

#### *Scenario alternativo*: nome utente o password scorretta

- Se l'autorizzazione fallisce, l'utente ritorna allo schermo iniziale (**index.jsp**) in cui viene visualizzato un messaggio di errore

### “Utente effettua rilancio”

#### *Scenario Principale*

- L'utente accede ad uno schermo (**offerta.jsp**) nel quale sono riassunti i dati dell'utente (nome, cognome). Lo schermo contiene inoltre un elenco degli oggetti all'asta, con una maschera attraverso cui l'utente può effettuare un'offerta per uno degli oggetti (**NOTA**: si consiglia di prevedere un controllo di tipo “select” o “radio” per specificare l'oggetto per cui effettuare l'offerta). Gli oggetti all'asta sono memorizzati in una tabella **Oggetti** della base di dati che contiene il codice dell'oggetto, la descrizione, la migliore offerta ricevuta e il riferimento all'utente che ha effettuato l'offerta
- Il sistema convalida i dati forniti; se i dati sono corretti il sistema registra la nuova offerta da parte dell'utente (**NOTA**: la nuova offerta deve essere superiore alla precedente)
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i risultati dell'operazione (dati dell'oggetto e nuova offerta); lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**offerta.jsp**) nel caso l'utente voglia eseguire una ulteriore operazione; (b) un collegamento ad uno schermo di uscita (**uscita.jsp**) attraverso il quale l'utente può abbandonare l'applicazione

#### *Scenario alternativo*: offerta scorretta

- Se la convalida fallisce, l'utente ritorna allo schermo di immissione (**offerta.jsp**) in cui viene visualizzato un messaggio di errore

Scrivere l'applicazione Web che effettua le operazioni elencate sopra secondo le seguenti specifiche:

- L'applicazione deve utilizzare pagine JSP e un'architettura di tipo **Modello 1** oppure un'architettura di tipo **Modello 2** basata sull'utilizzo del framework didattico **pinco**
- E' necessario provvedere alla **convalida** dei dati sottomessi dall'utente
- La grafica deve essere organizzata utilizzando un foglio di stile **CSS**
- Non è necessario provvedere alla convalida del codice HTML prodotto dalle pagine JSP
- Per la gestione della base di dati è necessario utilizzare **PostgreSQL**, e il relativo driver **JDBC**

## Livello avanzato

E' necessario scrivere un'applicazione Web che consente di partecipare ad aste in linea. Le aste riguardano oggetti di vario tipo (es: macchine fotografiche, cellulari, automobili, ciclomotori). L'applicazione deve consentire di eseguire i seguenti caso d'uso.

### “Utente effettua login”

#### Scenario Principale

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per fornire nome utente e password
- Il sistema autentica l'utente accedendo ad una tabella **Utenti** della base di dati, che contiene il nome utente, la password, il nome e cognome (**NOTA**: è necessario prevedere tra gli altri un utente “Mario Rossi” con nome utente **utente** e password **utente**)
- Se l'autenticazione ha successo, l'utente accede allo schermo iniziale del caso d'uso successivo (**iniziale.jsp**)

Scenario alternativo: nome utente o password scorretta

- Se l'autorizzazione fallisce, l'utente ritorna allo schermo iniziale (**index.jsp**) in cui viene visualizzato un messaggio di errore

### “Utente effettua rilancio”

#### Scenario Principale

- L'utente accede ad uno schermo (**iniziale.jsp**) nel quale sono riassunti i dati dell'utente (nome, cognome). Lo schermo contiene inoltre un elenco degli oggetti all'asta con la relativa migliore offerta ricevuta fino a quel punto; l'elenco deve prevedere una maschera attraverso cui l'utente può selezionare uno degli oggetti per cui fare un'offerta (**NOTA**: si consiglia di prevedere un controllo di tipo “select” o “radio” per specificare l'oggetto per cui effettuare l'offerta). Gli oggetti all'asta sono memorizzati in una tabella **Oggetti** della base di dati che contiene il codice dell'oggetto, la descrizione, la migliore offerta ricevuta fino a quel punto e il riferimento all'utente che ha effettuato l'offerta
- Selezionando l'oggetto, l'utente accede ad uno schermo (**offerta.jsp**) in cui sono riassunti i dati dell'oggetto selezionato – codice, descrizione e migliore offerta fino a quel punto – con una maschera per effettuare una nuova offerta; attraverso la maschera l'utente fornisce (a) l'importo della nuova offerta; (b) il proprio numero di carta di credito (per semplicità di 6 cifre)
- Il sistema convalida i dati forniti; se i dati sono corretti il sistema registra la nuova offerta da parte dell'utente in una tabella **Offerte** della base di dati che contiene l'importo dell'offerta, il numero di carta di credito, il riferimento all'utente che ha effettuato l'offerta e all'oggetto per cui è stata fatta l'offerta; inoltre, il sistema aggiorna la migliore offerta per l'oggetto in questione (**NOTA**: la nuova offerta deve essere superiore alla precedente)
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i risultati dell'operazione (dati dell'oggetto e nuova offerta effettuata); lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**iniziale.jsp**) nel caso l'utente voglia eseguire una ulteriore operazione; nel caso l'utente effettui diversi rilanci, durante la sessione di lavoro è necessario visualizzare sullo schermo tutti i rilanci effettuati (b) un collegamento ad uno schermo di uscita (**uscita.jsp**) attraverso il quale l'utente può abbandonare l'applicazione

Scenario alternativo: offerta scorretta

- Se la convalida fallisce, l'utente ritorna allo schermo di immissione (**offerta.jsp**) in cui viene visualizzato un messaggio di errore

Scrivere l'applicazione Web che effettua le operazioni elencate sopra secondo le seguenti specifiche:

- L'applicazione deve utilizzare pagine JSP e un'architettura di tipo **Modello 2** basata sul framework didattico **pinco**, ed uno strato di persistenza basato su DAO e gestione delle transazioni
- E' possibile scegliere se utilizzare o meno un pool di connessioni
- E' necessario provvedere alla **convalida** dei dati sottomessi dall'utente
- Il codice HTML prodotto deve essere conforme al **DTD XHTML-Basic**. E' necessario procedere alla convalida del codice prodotto utilizzando il validatore **xmlvalid**
- La grafica deve essere organizzata utilizzando un foglio di stile **CSS**.
- Per la gestione della base di dati è necessario utilizzare **PostgreSQL**, e il relativo driver **JDBC**

## Prova d'Esame

Cognome e Nome \_\_\_\_\_ Matricola \_\_\_\_\_

Quesito svolto base ?      intermedio ?      avanzato ?

Tempo a Disposizione: 4 ore e mezza

## Scommesse

### Livello base

E' necessario scrivere un'applicazione Web che consente di effettuare scommesse sui risultati delle gare dei 100 mt. di atletica alle Olimpiadi. L'applicazione deve consentire di eseguire il seguente caso d'uso.

#### “Utente aggiorna quota corridore”

##### *Scenario Principale*

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per specificare i dati del corridore di cui bisogna aggiornare la quota. Nella maschera è necessario specificare il nome del corridore (es: Tim Montgomery, Maurice Greene ecc.)
- Il sistema verifica che nella base di dati esista il corridore specificato. I dati delle quote sono memorizzati in una tabella **Quote** della base di dati che contiene il nome del corridore, l'importo totale delle scommesse ricevute per il corridore fino a quel punto e la quota, che per semplicità può essere considerata una stringa (es: “2 a 1”, oppure “15 a 2” ecc.)
- Se il corridore esiste, l'utente accede ad uno schermo (**aggiornamento.jsp**) nel quale sono riassunti i dati del corridore; lo schermo contiene inoltre una maschera attraverso la quale l'utente fornisce la quota aggiornata
- Il sistema aggiorna la quota per il corridore
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale viene riassunta la quota aggiornata; lo schermo contiene un collegamento allo schermo iniziale nel caso l'utente voglia eseguire una ulteriore operazione

##### *Scenario alternativo: il corridore non esiste*

- L'utente accede ad uno schermo (**errore.jsp**) in cui viene visualizzato un messaggio di errore ed un collegamento allo schermo principale per effettuare una nuova operazione

Scrivere l'applicazione Web che effettua le operazioni elencate sopra secondo le seguenti specifiche:

- L'applicazione deve utilizzare pagine JSP e un'architettura di tipo **Modello 1**.
- La grafica deve essere organizzata utilizzando un foglio di stile **CSS**.
- Non è necessario provvedere alla convalida dei dati sottomessi dall'utente.
- Non è necessario provvedere alla convalida del codice HTML prodotto dalle pagine JSP.

## Livello intermedio

E' necessario scrivere un'applicazione Web che consente di effettuare scommesse sui risultati delle gare dei 100 mt. di atletica alle Olimpiadi. L'applicazione deve consentire di eseguire i seguenti caso d'uso.

### “Utente effettua login”

#### *Scenario Principale*

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per fornire nome utente e password
- Il sistema autentica l'utente accedendo ad una tabella **Utenti** della base di dati, che contiene il nome utente, la password, il nome e cognome dell'utente (**NOTA:** è necessario prevedere tra gli altri un utente “Mario Rossi” con nome utente **utente** e password **utente**)
- Se l'autenticazione ha successo, l'utente accede allo schermo principale del caso d'uso successivo (**aggiornamento.jsp**)

*Scenario alternativo:* nome utente o password scorretta

- Se l'autorizzazione fallisce, l'utente ritorna allo schermo iniziale (**index.jsp**) in cui viene visualizzato un messaggio di errore

### “Utente aggiorna quota corridore”

#### *Scenario Principale*

- L'utente accede ad uno schermo (**aggiornamento.jsp**) nel quale sono riassunti i dati dell'utente (nome e cognome). Lo schermo contiene inoltre una maschera che elenca i corridori per cui si accettano scommesse (es: Tim Montgomery, Maurice Greene, ecc.) con la relative quota attuale (**NOTA:** per semplicità si supponga che la quota sia fatta di due numeri interi, es: 2 a 1, 15 a 2 ecc.). La maschera deve consentire di aggiornare la quota di uno di loro (**NOTA:** si consiglia di prevedere un controllo di tipo “select” o “radio” per specificare il corridore per cui aggiornare le quote). I dati dei corridori sono memorizzati in una tabella **Corridori** della base di dati che contiene il nome del corridore, la quota attuale ed un riferimento all'utente che ha inserito la quota
- Il sistema convalida i dati forniti; se i dati sono corretti aggiorna la quota del corridore
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i dati del corridore e la nuova quota; lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**aggiornamento.jsp**) nel caso l'utente voglia eseguire una ulteriore operazione; (b) un collegamento ad uno schermo di uscita (**uscita.jsp**) attraverso il quale l'utente può abbandonare l'applicazione

*Scenario alternativo:* dati scorretti della quota

- Se la convalida dei dati fallisce, l'utente ritorna allo schermo iniziale (**aggiornamento.jsp**) in cui viene visualizzato un messaggio di errore

Scrivere l'applicazione Web che effettua le operazioni elencate sopra secondo le seguenti specifiche:

- L'applicazione deve utilizzare pagine JSP e un'architettura di tipo **Modello 1** oppure un'architettura di tipo **Modello 2** basata sull'utilizzo del framework didattico **pinco**
- E' necessario provvedere alla **convalida** dei dati sottomessi dall'utente
- La grafica deve essere organizzata utilizzando un foglio di stile **CSS**
- Non è necessario provvedere alla convalida del codice HTML prodotto dalle pagine JSP
- Per la gestione della base di dati è necessario utilizzare **PostgreSQL**, e il relativo driver **JDBC**

## Livello avanzato

E' necessario scrivere un'applicazione Web che consente di effettuare scommesse sui risultati delle gare dei 100 mt. di atletica alle Olimpiadi. L'applicazione deve consentire di eseguire i seguenti caso d'uso.

### “Utente effettua login”

#### Scenario Principale

- L'utente accede ad uno schermo iniziale (**index.jsp**) nel quale viene visualizzata una maschera per fornire nome utente e password
- Il sistema autentica l'utente accedendo ad una tabella **Utenti** della base di dati, che contiene il nome utente, la password, il nome e cognome dell'utente (**NOTA**: è necessario prevedere tra gli altri un utente “Mario Rossi” con nome utente **utente** e password **utente**)
- Se l'autenticazione ha successo, l'utente accede allo schermo principale del caso d'uso successivo (**iniziale.jsp**)

Scenario alternativo: nome utente o password scorretta

- Se l'autorizzazione fallisce, l'utente ritorna allo schermo iniziale (**index.jsp**) in cui viene visualizzato un messaggio di errore

### “Utente effettua scommessa”

#### Scenario Principale

- L'utente accede ad uno schermo (**iniziale.jsp**) nel quale sono riassunti i dati dell'utente (nome, cognome). Lo schermo contiene inoltre un elenco dei corridori per cui sono accettate scommesse (es: Tim Montgomery, Maurice Greene, ecc.) con le relative quote e l'importo totale delle scommesse ricevute fino a quel punto per il corridore (**NOTA**: per semplicità si supponga che la quota sia fatta di due numeri interi, es: 2 a 1, 15 a 2 ecc.); l'elenco deve prevedere una maschera attraverso cui l'utente può selezionare uno dei corridori su cui scommettere (**NOTA**: si consiglia di prevedere un controllo di tipo “select” o “radio” per specificare il corridore su cui scommettere). I corridori sono memorizzati in una tabella **Corridori** della base di dati che contiene il codice del corridore, il nome del corridore, la sua quota e l'importo complessivo delle scommesse ricevute per il corridore fino a quel punto
- Selezionando il corridore, l'utente accede ad uno schermo (**scommessa.jsp**) in cui sono riassunti i dati del corridore selezionato – nome e quota – con una maschera per effettuare la scommessa; attraverso la maschera l'utente fornisce (a) l'importo della scommessa; (b) il proprio numero di carta di credito (per semplicità di 6 cifre)
- Il sistema convalida i dati forniti; se i dati sono corretti il sistema registra la nuova scommessa da parte dell'utente in una tabella **Scommesse** della base di dati che contiene l'importo della scommessa, il numero di carta di credito, il riferimento all'utente che ha effettuato la scommessa e al corridore su cui è stata fatta la scommessa; il sistema aggiorna inoltre l'importo totale delle scommesse ricevute per il corridore (**NOTA**: la scommessa minima ammessa è di 10 Euro)
- L'utente accede ad uno schermo (**conferma.jsp**) nel quale vengono riassunti i risultati dell'operazione (dati del corridore e della scommessa); lo schermo contiene inoltre (a) un collegamento allo schermo iniziale del caso d'uso (**iniziale.jsp**) nel caso l'utente voglia eseguire una ulteriore scommessa; nel caso in cui l'utente effettui diverse scommesse, durante la sessione di lavoro è necessario visualizzare sullo schermo tutte le scommesse effettuate; (b) un collegamento ad uno schermo di uscita (**uscita.jsp**) attraverso il quale l'utente può abbandonare l'applicazione

Scenario alternativo: dati scorretti della scommessa

- Se la convalida fallisce, l'utente ritorna allo schermo di immissione (**scommessa.jsp**) in cui vengono visualizzati i necessari messaggi di errore

Scrivere l'applicazione Web che effettua le operazioni elencate sopra secondo le seguenti specifiche:

- L'applicazione deve utilizzare pagine JSP e un'architettura di tipo **Modello 2** basata sul framework didattico **pinco**, ed uno strato di persistenza basato su DAO e gestione delle transazioni
- E' possibile scegliere se utilizzare o meno un pool di connessioni
- E' necessario provvedere alla **convalida** dei dati sottomessi dall'utente
- Il codice HTML prodotto deve essere conforme al **DTD XHTML-Basic**. E' necessario procedere alla convalida del codice prodotto utilizzando il validatore **xmllvalid**
- La grafica deve essere organizzata utilizzando un foglio di stile **CSS**.
- Per la gestione della base di dati è necessario utilizzare **PostgreSQL**, e il relativo driver **JDBC**