

PROGETTO DI PROGRAMMAZIONE AD OGGETTI

Marco Tesser, matricola 1201154

Emma Roveroni, matricola 1187275

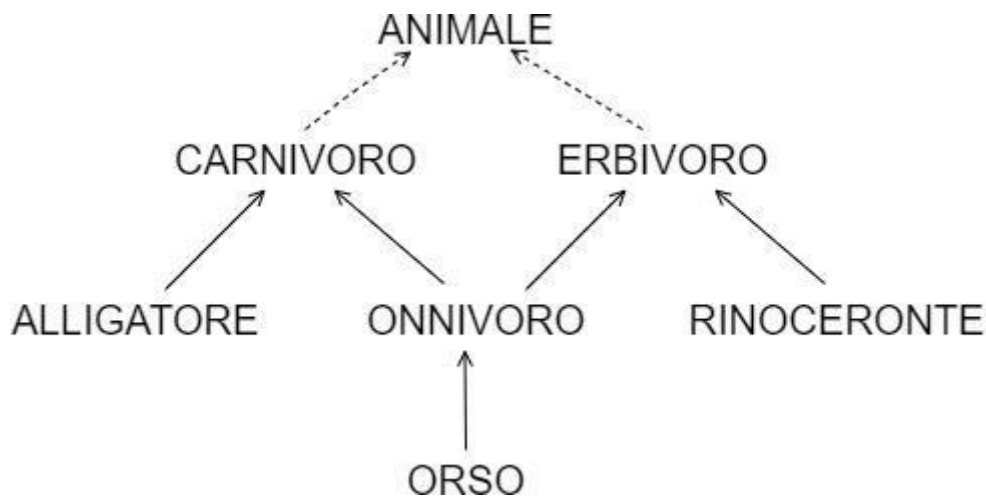
INTRODUZIONE

Qanimals è un'applicazione gestionale che permette l'organizzazione della compravendita di animali esotici e l'amministrazione delle finanze relative ad essa. Vengono offerte tutte le funzioni necessarie all'utente per poter organizzare il suo business quali : inserimento , rimozione , caricamento e salvataggio di dati relativi ad animali , acquisti e vendite con l'ausilio di un contenitore per poter salvare tutte queste informazioni.

GERARCHIA

La gerarchia serve per rappresentare tutti i possibili animali esotici che possono essere salvati nel contenitore e consiste in:

- Una classe base astratta "Animale" che rappresenta l'animale nella sua forma più generica
- Una classe astratta "Erbivoro" derivata virtualmente da "Animale".
- Una classe astratta "Carnivoro" derivata virtualmente da "Animale".
- Una classe astratta "Onnivoro" derivata da "Erbivoro" e "Carnivoro".
- Una classe concreta "Rinoceronte" derivata da "Erbivoro".
- Una classe concreta "Alligatore" derivata da "Carnivoro".
- Una classe concreta "Orso" derivata da "Onnivoro".



ANIMALE:

La classe base della gerarchia rappresenta l'animale a livello più generico possibile e fornisce quindi campi dati comuni a qualsiasi tipo di animale come: l'ID , il suo peso , altezza , lunghezza ed età , il fatto che sia in estinzione o meno e la sede dove è tenuto.

ERBIVORO:

Questa classe rappresenta un animale erbivoro e lo caratterizza in base alla lista di alimenti di cui può nutrirsi.

CARNIVORO:

Questa classe rappresenta un animale carnivoro e lo caratterizza in base al fatto di essere velenoso o necrofago o meno.

ONNIVORO:

Questa classe rappresenta un animale onnivoro.

RINOCERONTE:

Questa classe rappresenta un rinoceronte e lo caratterizza in base al peso in kg delle sua corna.

ALLIGATORE:

Questa classe rappresenta un alligatore e lo caratterizza in base al peso in kg della sua pelle.

ORSO:

Questa classe rappresenta un orso e lo caratterizza in base al peso in kg delle sua pelliccia.

CHIAMATE POLIMORFE

La classe base animale implementa 6 metodi virtuali puri:

1. isGrasso() per determinare se l'animale è grasso o meno.
2. isVecchio() per determinare se l'animale è vecchio o meno.
3. isDiValore() per determinare se l'animale è di valore o meno.
4. GetTipo() per determinare il tipo di animale (Erbivoro , Carnivoro, Onnivoro)
5. GetSpecie() per determinare la specie di animale (Rinoceronte , Alligatore , Orso)
6. Clone () per clonare gli animali della gerarchia

INPUT/OUTPUT

Il progetto consente il caricamento e salvataggio delle informazioni relative al contenitore. Si è scelto di adottare l'uso dei file xml come formato di file per la persistenza.

Sono state implementati i metodi load e save che fanno uso delle classi QXmlStreamWriter

e QXmlStreamReader per la codifica e decodifica del file xml.

Il metodo save viene chiamato manualmente dall'utente ogni volta che desidera salvare il lavoro completato invece load viene chiamato all'apertura dell'applicazione per caricare i dati salvati in data.xml

MANUALE GUI

Per lo sviluppo della parte relativa alla GUI è stato adottato il pattern Model-Controller-View.

La parte grafica si compone di una MainWindow principale che viene lanciata all'avvio dell'applicazione, la MainWindow è poi a sua volta suddivisa in due parti : la parte di inventario che mostra tutti gli animali salvati nel contenitore e permette l'aggiunta e rimozione di essi e la parte finanziaria che registra tutti gli acquisti , vendite e bilancio relative agli animali.

Per aggiungere un animale all'inventario basta premere il pulsante "Aggiungi animale" in fondo alla pagina che aprirà una finestra dove inserire i dati dell'animale e , se l'animale è stato acquistato , si potranno aggiungere i dati dell'acquisto che successivamente verranno aggiunti automatico nella parte finanziaria nella sezione "Acquisti" ed il bilancio aggiornato.

Per rimuovere un animale basta premere il pulsante di delete alla destra degli attributi nella tabella dell'inventario. A questo si può scegliere se semplicemente eliminare l'animale dall'inventario oppure se aggiungerlo anche alla lista degli animali venduti , se si sceglie la seconda opzione allora comparirà un'ulteriore finestra che chiederà di aggiungere i dati relativi alla vendita che verranno poi aggiunti in automatico nella parte finanziaria nella sezione "Vendite" ed il bilancio aggiornato.

Per visualizzare info aggiuntive riguardanti un animale specifico basta premere il pulsante di info alla destra degli attributi nella tabella dell'inventario.

Se si ha sbagliato ad inserire i dati relativi ad un acquisto esso potrà essere rimosso direttamente dalla lista acquisti ed il bilancio aggiornato in automatico , di conseguenza si andrà ad eliminare l'animale relativo all'acquisto nella parte di inventario e i dati potranno essere inseriti nuovamente. Il medesimo discorso si applica per vendita.

Per salvare i progressi e poi uscire dall'applicazione basta premere il pulsante "Menu" in alto a sinistra che aprirà un menù a tendina con le opzioni di "Save" ed "Exit".

RIPARTIZIONE ORE

La realizzazione del progetto è durata circa 50 ore ripartite nel seguente modo:

1. Analisi preliminare del problema: 1 h
2. Progettazione modello e GUI: 5 h
3. Apprendimento libreria Qt: 5 h
4. Codifica modello e GUI : 30 h
5. Debugging : 8 h
6. Testing : 1 h

SUDDIVISIONE DEL LAVORO PROGETTUALE

- Modello:
 - Gerarchia : Marco Tesser , Emma Roveroni
 - Contenitore : Marco Tesser , Emma Roveroni
 - Dati finanziari : Emma Roveroni
- Controller : Marco Tesser , Emma Roveroni
- GUI :
 - Vista : Marco Tesser , Emma Roveroni
 - Vista Lista : Marco Tesser
 - Vista Lista Add : Marco Tesser
 - Vista Lista Options : Marco Tesser
 - Vista Lista Vendita : Marco Tesser
 - Vista Finanza : Emma Roveroni
 - Vista Finanza Acquisti : Emma Roveroni
 - Vista Finanza Vendite : Emma Roveroni
 - Vista Finanza Bilancio : Emma Roveroni
 - Vista Add Button : Marco Tesser
- Save/Load : Marco Tesser , Emma Roveroni

AMBIENTE DI SVILUPPO

Il sistema operativo utilizzato per la realizzazione del progetto è Windows 10 64bit. La versione Qt utilizzata è Qt 5.9.9. Il compilatore utilizzato è MinGW 7.3.0 64bit.