

Präsenz 2

Für die Aufgaben benötigt ihr die Dateien aus der Tierverwaltung.jar aus dem moodle. Mache dich mit dem Code vertraut.

Aufgabe 1

1. Implementiere ein Menue in der Datei `Menue` (view) für die Verwaltung unserer Tierwelt. Unter dem Menüpunkt **FILE** sind die Menüeltems **LOAD**, **SAVE** und **EXIT** zu finden. Unter **LOAD** und **SAVE** befindet sich eine Aufteilung **TIER** und **PERSON**. Ein weiterer Menüpunkt ist **EDIT** und darunter die Menüeltems **INSERT**, **DELETE** und **CHANGE**. **INSERT** die auch aufgeteilt in **TIER** und **PERSON**. Ein dritter Menüpunkt soll eine Möglichkeit zum Sortieren (**SORT**) der Tiere (**TIER**) und Personen (**PERSON**) geben.
2. Belege die Menüpunkte mit folgenden Funktionen (`setOnAction`):
 - LOAD**: Lesen aus einer Tier bzw. Persondatei und auf dem Hauptfenster anzeigen. Die Aktion übernimmt der `LoadController`.
 - SAVE**: Speicher der Tier- bzw. Personliste (bevor das erste Mal eine Datei geladen werden kann muss die entsprechende DefaultListe in eine Datei geschrieben werden). Die Aktion übernimmt der `SaveController`.
 - EXIT**: Beenden des Programms (`close()`)
 - INSERT -> TIER**: Öffnet ein neues Fenster zum Einfügen eines Tieres mit Auswahl der **KLASSE** (`RadioButton`), Eingabe der **ART** und **ALTER**. Handelt es sich um ein Haustier (Hund oder Vogel), erscheint nach Auswahl dieser Klassen ein weiteres Eingabefeld **NAME**. Zwei Buttons sind ersichtlich: **SCHLIEßEN** (schließt das aktuelle Fenster) und **SUBMIT** (schreibt in die entsprechende Liste und zeigt die neue Liste im Hauptfenster an). Das Öffnen des Dialogfensters übernimmt der `InsertDialogController`.
 - INSERT -> PERSON**: Öffnet ein neues Fenster zum Einfügen einer Person mit **VORNAME**, **NACHNAME**, Auswahl des **GESCHLECHTS** (`RadioButton`), **ALTER**, **EMAIL** und **BILD**. Zusätzlich gibt es die Auswahl, ob die Person ein Haustier hat oder nicht (`RadioButton`). Falls ein Haustier vorhanden ist, dann öffnet sich ein weiteres Fenster für die Eingabe des Tieres mit der Auswahl **HUND** oder **VOGEL** (`RadioButton`) und die Eingabefelder **ART**, **ALTER** und **NAME**. Zwei Buttons sind ersichtlich: **SCHLIEßEN** (schließt das aktuelle Fenster) und **SUBMIT** (schreibt in die entsprechende Liste und zeigt die neue Liste im Hauptfenster an). Das Öffnen des Dialogfensters übernimmt der `InsertDialogController`.

Die Menüpunkte **DELETE** und **CHANGE** sind mit keiner Aktion belegt.

SORT -> TIER: Sortiert die Liste der Tiere nach einem beliebigen Kriterium. Die sortierte Liste wird direkt angezeigt. Die Aktion übernimmt der `SortController`.

SORT -> PERSON: Sortiert die Liste der Personen nach Vornamen. Die sortierte Liste wird direkt angezeigt. Die Aktion übernimmt der `SortController`.

Für die einzelnen Aktionen müssen die entsprechende Tier - bzw. Personliste geladen und im Hauptfenster ersichtlich sein. Die Unterscheidung zwischen Tier und Person in den Controllern wird mithilfe eines booleans `person` getroffen.

Aufgabe 2

1. Implementiere die Methode `handle(ActionEvent e)` des Controllers `InsertTierController`. Das neue Tier wird der Tierliste hinzugefügt (`addTier` in `Tiere`). Beachte, dass für die Erstellung eines Tierobjektes die Tierklasse bekannt sein muss. Die Eingabefelder werden geleert.
 2. Ergänze die Klasse `DialogTier` (s.ToDo). Der Name des Tieres soll nur erscheinen, wenn das Tier ein Haustier ist. Die Entscheidung ob ein Haustier oder nicht wird über das boolean `home` (s. Methodenkopf) geregelt. Der Parameter muss über den Controller `InsertDialogController` übergeben werden.
 3. Ergänze die Klasse `DialogPerson` (s.ToDo). Wenn die Person ein Hasutier hat, soll sich der Dialog für die Eingabe des Tieres öffnen (über `InsertDialogController`). In dem Dialog erscheinen dann nur die Haustiere Hund und Vogel. Der Parameter `home` muss über den Controller `InsertDialogController` übergeben werden.
 4. Ergänze im `SortController` die Sortierung der Tiere nach einem beliebigen Kriterium.
-