

Dokumentation der Klasse KartenStart

Die Klasse dient zum Starten der Anwendung.

Auftrag	public static void main(String args[]) Die Anwendung wird gestartet und ein StartFenster wird erzeugt.
---------	---

Dokumentation der Klasse Hauptansicht

Die Klasse dient zur Realisierung der Ansicht der Karteiverwaltungsanwendung.

Konstruktor	Hauptansicht(List listeVonStart) Es wird ein Fenster erstellt, in dem die Daten einer Karteikarte angezeigt werden können, sowie Schaltflächen für verschiedene Handlungsoptionen zur Verfügung stehen: Weiterblättern, neue Karte erstellen, zur ersten Karte wechseln, Liste der Karten speichern, Karte suchen, Karte löschen, Karte ändern. Diese Optionen beziehen sich immer auf die im Fenster angezeigte aktuelle Karte von Karten in einer Liste. Diese Liste wird übergeben und unter dem Namen „lis“ gespeichert. Anfangs ist die erste Karte der Liste zu sehen, falls sie existiert, bzw. ein Hinweis, dass kein Zugriff möglich ist, wenn die Liste leer ist. Außerdem werden Fenster zur Veränderung einer Karte, zum Suchen einer Karte.
Auftrag	void setKarte() Die aktuelle Karte der Liste ist im Fenster zu sehen, falls sie existiert, bzw. ein Hinweis, dass kein Zugriff möglich ist, wenn die Liste leer ist.
Auftrag	void zeigeAenderFenster() Ein Fenster zur Veränderung der aktuellen Karte wird sichtbar.
Auftrag	void versteckeAenderFenster () Ein Fenster zur Veränderung der aktuellen Karte wird unsichtbar.
Auftrag	void zeigeSuchFenster () Ein Fenster zur Suche einer Karte wird sichtbar.
Auftrag	void versteckeSuchFenster() Ein Fenster zur Suche einer Karte wird unsichtbar.
Auftrag	void zeigeNeuFenster() Ein Fenster zur Eingabe der Daten für eine neue Karte wird sichtbar.
Auftrag	void versteckeNeuFenster() Ein Fenster zur Eingabe der Daten für eine neue Karte wird unsichtbar.
Auftrag	void zeigeWirklichLoeschFenster () Ein Fenster zur Abfrage, ob die aktuelle Karte mit ihren Daten wirklich gelöscht werden soll, wird sichtbar.
Auftrag	void versteckeWirklichLoeschFenster() Ein Fenster zur Abfrage, ob die aktuelle Karte mit ihren Daten wirklich gelöscht werden soll, wird unsichtbar.
Auftrag	void macheSichtbar() Das Hauptansichtsfenster wird sichtbar.
Auftrag	void macheUnsichtbar() Das Hauptansichtsfenster wird unsichtbar.
Auftrag	void nimmListe(List li) Die übergebene Liste wird als Liste „lis“ der Karteikarten gespeichert.

Dokumentation der Klasse HauptKlickEmpfaenger

Die Klasse dient ermöglicht die Reaktion auf die Benutzereingaben im Hauptfenster.

Konstruktor	HauptKlickEmpfaenger(Hauptansicht eingabe, List l) Das übergebene Hauptansichtsfenster wird unter dem Namen „haupt“ gespeichert, die übergebene Liste unter dem Namen „lis“.
Auftrag	void actionPerformed(ActionEvent e) Die Methode bewirkt, dass bei einem Klick auf eine Schaltfläche eine der Aufschrift "neue Karte", „Karte suchen“, „Karte ändern“, „Karte löschen“ das entsprechende Fenster sichtbar gemacht wird. Bei einem Klick auf die Schaltfläche „weiter blättern“, soll in der Liste eine Karteikarte „weitergeblättert werden“, d.h. das auf die aktuelle Karte folgende Element wird das aktuelle Element und im Hauptfenster angezeigt, falls es existiert, ansonsten gibt es dann kein aktuelles Element und im Hauptfenster erscheint die Meldung „Momentan kein Zugriff auf eine Karte!“. Bei einem Klick auf die Schaltfläche „Zur ersten Karte“ wird ggf. die erste Karte die aktuelle und im Hauptfenster angezeigt. Bei einem Klick auf die Schaltfläche mit der Aufschrift „Liste speichern“ wird die Methode „listeSpeichern()“ zum Speichern der Liste ausgeführt.
Auftrag	void void listeSpeichern() Die Liste wird in der Datei „Kartei.dat“ gespeichert.

Dokumentation der Klasse AenderFenster

Die Klasse dient zur Realisierung der Ansicht des Fensters zur Veränderung einer Karte.

Konstruktor	AenderFenster(List l,Hauptansicht h) Es wird ein Fenster erstellt, in dem neben den alten Daten einer Karteikarte neue eingegeben werden können. Eine Schaltfläche zum Übernehmen der Änderungen steht ebenfalls zur Verfügung.
Anfrage	String getAltenName() Der bisher gespeicherte Name (aus dem entsprechenden Label) wird zurückgegeben.
Anfrage	String getAltenWohnort() Der bisher gespeicherte Wohnort (aus dem entsprechenden Label) wird zurückgegeben.
Anfrage	int getAltePLZ() Die bisher gespeicherte Postleitzahl (aus dem entsprechenden Label) wird zurückgegeben.
Anfrage	String getAlteStrasse() Die bisher gespeicherte Strasse (aus dem entsprechenden Label) wird zurückgegeben.
Anfrage	String getNeuenName() Die im entsprechenden Textfeld als neuer Name eingegebene Zeichenkette wird zurückgegeben.
Anfrage	String getNeuenWohnort() Die im entsprechenden Textfeld als neuer Wohnort eingegebene Zeichenkette wird zurückgegeben.
Anfrage	int getNeuePLZ()

	Die im entsprechenden Textfeld als neue Postleitzahl eingegebene Zahl wird zurückgegeben.
Anfrage	String getNeueStrasse() Die im entsprechenden Textfeld als neue Straße eingegebene Zeichenkette wird zurückgegeben.
Auftrag	void setAltenName(String na) Der Name „na“ wird als bisheriger Name im entsprechenden Label angezeigt.
Auftrag	void setAltenWohnort(String w) Der Wohnort „w“ wird als bisheriger Wohnort im entsprechenden Label angezeigt.
Auftrag	void setAltePLZ(int i) Der übergebene Zahl wird als bisherige Postleitzahl im entsprechenden Label angezeigt.
Auftrag	void setAlteStrasse(String str) Die Straße „str“ wird als bisherige Strasse im entsprechenden Label angezeigt.
Auftrag	void aenderFelderloeschen() Die Textfelder zum Verändern der Daten werden geleert.
Auftrag	void macheSichtbar() Das Fenster wird sichtbar.
Auftrag	void macheUnsichtbar() Das Fenster wird unsichtbar.
Auftrag	void setKarte() Falls es eine aktuelle Karte gibt, werden die Daten als Daten der zu verändernden Karte in dem Fenster angezeigt.

Dokumentation der Klasse AenderKlickEmpfaenger

Die Klasse dient ermöglicht die Reaktion auf die Benutzereingaben im Fenster zur Veränderung der Daten auf einer Karteikarte.

Konstruktor	AenderKlickEmpfaenger(AenderFenster neu, List l, Hauptansicht h) Das übergebene Hauptansichtsfenster wird unter dem Namen „haupt“ gespeichert, die übergebene Liste unter dem Namen „lis“, das AenderFenster unter „af“.
Auftrag	void actionPerformed(ActionEvent e) Durch Klicken auf die Schaltfläche „Änderung übernehmen“ werden die Änderungen der Daten auf der Karte in der Liste übernommen. Das AenderFenster zur Veränderung der Daten einer Karteikarte wird unsichtbar gemacht. Im Hauptfenster ist die veränderte Karte zu sehen.

Dokumentation der Klasse WirklichLoeschFenster

Die Klasse dient zur Realisierung der Ansicht des Fensters zum Löschen einer Karte.

Konstruktor	WirklichLoeschFenster(List l, Hauptansicht h) Es wird ein Fenster erstellt, in dem die Daten der aktuellen Karteikarte angezeigt werden sowie eine Schaltfläche zum Löschen der Karte.
Anfrage	String getName()

	Der bisher gespeicherte Name (aus dem entsprechenden Label) wird zurückgegeben.
Anfrage	String getWohnort() Der bisher gespeicherte Wohnort (aus dem entsprechenden Label) wird zurückgegeben.
Anfrage	int getPLZ() Die bisher gespeicherte Postleitzahl (aus dem entsprechenden Label) wird zurückgegeben.
Anfrage	String getStrasse() Die bisher gespeicherte Straße (aus dem entsprechenden Label) wird zurückgegeben.
Auftrag	void setName(String na) Der Name „na“ wird als bisheriger Name im entsprechenden Label angezeigt.
Auftrag	void setWohnort(String w) Der Wohnort „w“ wird als bisheriger Wohnort im entsprechenden Label angezeigt.
Auftrag	void setPLZ(int i) Der übergebene Zahl wird als bisherige Postleitzahl im entsprechenden Label angezeigt.
Auftrag	void setStrasse(String str) Die Straße „str“ wird als bisherige Straße im entsprechenden Label angezeigt.
Auftrag	void macheSichtbar() Das Fenster wird sichtbar.
Auftrag	void macheUnsichtbar() Das Fenster wird unsichtbar.

Dokumentation der Klasse LoeschKlickEmpfaenger

Die Klasse dient ermöglicht die Reaktion auf die Benutzereingaben im Fenster zur Veränderung der Daten auf einer Karteikarte.

Konstruktor	LoeschKlickEmpfaenger(WirklichLoeschFenster loe, List l, Hauptansicht h) Das übergebene Hauptansichtsfenster wird unter dem Namen „haupt“ gespeichert, die übergebene Liste unter dem Namen „lis“, das WirklichLoeschFenster unter „lf“.
Auftrag	void actionPerformed(ActionEvent e) Durch Drücken der Schaltfläche „endgültig löschen“ wird die aktuelle Karte aus der Liste gelöscht wird. Das Fenster zum Löschen wird unsichtbar gemacht. In der Hauptansicht soll dann die folgende Karte erscheinen (falls es eine solche gibt, ansonsten: Namen „Momentan kein Zugriff auf eine Karte!“).

Dokumentation der Klasse SuchFenster

Die Klasse dient zur Realisierung der Ansicht des Fensters zum Suchen einer Karte.

Konstruktor	SuchFenster(List l,Hauptansicht h) Es wird ein Fenster erstellt, in dem die Daten der aktuellen Karteikarte angezeigt werden sowie eine Schaltfläche zum Löschen der Karte. Das übergebene Hauptansichtsfenster wird unter dem Namen „haupt“ gespeichert, die übergebene Liste unter dem Namen „lis“.
-------------	--

Anfrage	String getName() Der eingegebene Name (aus dem entsprechenden Textfeld) wird zurückgegeben.
Anfrage	String getWohnort() Der eingegebene Wohnort (aus dem entsprechenden Textfeld) wird zurückgegeben.
Anfrage	int getPLZ() Die eingegebene Postleitzahl (aus dem entsprechenden Textfeld) wird als Zahl zurückgegeben. Wenn nichts eingegeben wurde wird 0 zurückgegeben.
Anfrage	String getStrasse() Die eingegebene Strasse (aus dem entsprechenden Textfeld) wird zurückgegeben.
Auftrag	void setName(String na) Der Name „na“ wird als Name im entsprechenden Textfeld angezeigt.
Auftrag	void setWohnort(String w) Der Wohnort „w“ wird als Wohnort im entsprechenden Textfeld angezeigt.
Auftrag	void setPLZ(int i) Der übergebene Zahl wird als Postleitzahl im entsprechenden Textfeld angezeigt.
Auftrag	void setStrasse(String str) Die Straße „str“ wird als Straße im entsprechenden Textfeld angezeigt.
Auftrag	void macheSichtbar() Das Fenster wird sichtbar.
Auftrag	void macheUnsichtbar() Das Fenster wird unsichtbar.

Dokumentation der Klasse SuchKlickEmpfaenger

Die Klasse dient ermöglicht die Reaktion auf die Benutzereingaben im Fenster zum Suchen einer Karteikarte.

Konstruktor	SuchKlickEmpfaenger(SuchFenster such, List l, Hauptansicht h) Das übergebene Hauptansichtsfenster wird unter dem Namen „haupt“ gespeichert, die übergebene Liste unter dem Namen „lis“, das SuchFenster unter „sf“.
Auftrag	void actionPerformed(ActionEvent e) Durch Drücken der Schaltfläche „Suche starten“ wird eine Karte gesucht, auf der der eingegebene Name, die eingegebene Straße, die eingegebene Postleitzahl oder der eingegebene Wohnort gespeichert ist. Das Fenster zum Suchen wird unsichtbar gemacht. Im Hauptfenster wird die erste Karte in der Liste angezeigt, bei der das erste angegebene Kriterium übereinstimmt. Falls es eine solche Karte nicht gibt, erscheint im Hauptfenster die Meldung „Momentan kein Zugriff auf eine Karte!“.

Dokumentation der Klasse NeuFenster

Die Klasse dient zur Realisierung der Ansicht des Fensters zum Erstellen einer Karte.

Konstruktor	NeuFenster(List l,Hauptansicht h) Es wird ein Fenster erstellt, in dem die Daten für eine neue Karteikarte eingegeben werden. Das übergebene Hauptansichtsfenster wird unter dem Namen „haupt“ gespeichert, die übergebene Liste unter dem Namen „lis“.
Anfrage	String getName() Der eingegebene Name (aus dem entsprechenden Textfeld) wird zurückgegeben.
Anfrage	String getWohnort() Der eingegebene Wohnort (aus dem entsprechenden Textfeld) wird zurückgegeben.
Anfrage	int getPLZ() Die eingegebene Postleitzahl (aus dem entsprechenden Textfeld) wird als Zahl zurückgegeben. Wenn nichts eingegeben wurde wird 0 zurückgegeben.
Anfrage	String getStrasse() Die eingegebene Straße (aus dem entsprechenden Textfeld) wird zurückgegeben.
Auftrag	void macheSichtbar() Das Fenster wird sichtbar.
Auftrag	void macheUnsichtbar() Das Fenster wird unsichtbar.

Dokumentation der Klasse NeuKlickEmpfaenger

Die Klasse dient ermöglicht die Reaktion auf die Benutzereingaben im Fenster zum Suchen einer Karteikarte.

Konstruktor	NeuKlickEmpfaenger(NeuFenster neu, List l,Hauptansicht h) Das übergebene Hauptansichtsfenster wird unter dem Namen „haupt“ gespeichert, die übergebene Liste unter dem Namen „lis“, das NeuFenster unter „nf“.
Auftrag	void actionPerformed(ActionEvent e) Durch Drücken der Schaltfläche „Neue Karte einfügen“ wird eine Karte erstellt, auf der der eingegebene Name, die eingegebene Straße, die eingegebene Postleitzahl oder der eingegebene Wohnort gespeichert ist. Sie wird in der Liste der Karten hinter der im Hauptfenster angezeigten Karte eingefügt und anschließend im Hauptfenster angezeigt. Das Fenster zum Erstellen einer Karte wird unsichtbar gemacht.

Dokumentation der Klasse StartFenster

Die Klasse dient zur Realisierung der Ansicht eines Fensters für eine Abfrage, ob eine Liste geladen werden soll oder eine neue Liste von Karteikarten erstellt werden soll.

Konstruktor	StartFenster() Es wird ein Fenster erstellt, in dem zwei Schaltflächen zu sehen sind, eine zum Laden einer Liste, eine zum Erstellen einer neuen
-------------	--

	Liste von Karteikarten.
Auftrag	void macheSichtbar() Das Fenster wird sichtbar.
Auftrag	void macheUnsichtbar() Das Fenster wird unsichtbar.

Dokumentation der Klasse StartKlickEmpfaenger

Die Klasse dient ermöglicht die Reaktion auf die Benutzereingaben im Fenster zum Suchen einer Karteikarte.

Konstruktor	StartKlickEmpfaenger(StartFenster sf) Das übergebene StartFenster unter dem Namen „start“ gespeichert.
Auftrag	void actionPerformed(ActionEvent e) Durch Drücken der Schaltfläche „neue Liste“ wird eine neue Liste lis erstellt. Ansonsten wird eine Liste aus der Datei „Kartei.dat“ geladen.
Auftrag	void listeLaden() Die Liste lis wird aus der Datei „Kartei.dat“ geladen.