

15_PhoneWithFileIO

Erstelle ein C#-Programm für die folgende Aufgabenstellung:

Wir möchten Kontakte in einem Telefonbuch verwalten. Dazu verwenden wir die Klassen **Contact** und **PhoneBook** und eine GUI, die uns die Benutzung dieser Klassen ermöglicht. Die Klasse **FileIO** ermöglicht uns das permanente Speichern der Kontaktdaten.

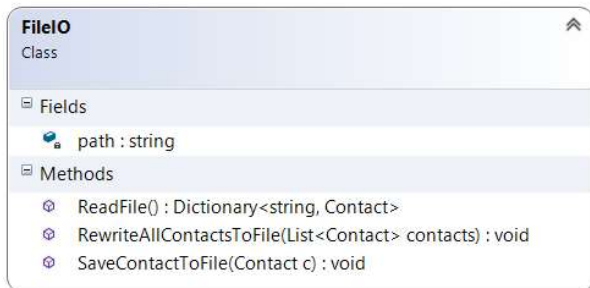
Klasse Contact

Die Klasse beschreibt ein **Contact**-Objekt mit seinem Namen, der Telefonnummer und mit **isFavorit**, ob es ein favorit contact ist oder nicht. Weiters besitzt ein Contact auch eine Historie aller Anrufe (**callHistory**). Dabei wird pro Anruf einfach nur die Anrufdauer in Minuten gespeichert. Die Anrufdauer muss eine positive Zahl sein und wird mit der Methode **Call** in der Liste **callHistory** gespeichert.

Name und Telefonnummer eines Contact-Object darf nicht leer sein. Verwende ExceptionHandling im Konstruktor um dies sicher zu stellen.

Achte darauf, dass bei der Übergabe nicht zulässiger Parameterwerte jeweils eine ArgumentException mit entsprechender Message geworfen werden muss!

Klasse FileIO



Diese Klasse implementiert die Operationen um die Daten der Anwendung auf die Festplatte speichern zu können bzw. diese von dort Einlesen zu können. Den Pfad zu Datei speichert das Feld **path**.

- **ReadFile** – Liest die Daten aus der Datei (**path** entsprechend), speichert die **Contact**-Objekte in eine **Dictionary** (Key ist der Name) und liefert diese als Rückgabewert.

Sollte das Dateiformat nicht in korrekt sein – also aus einer eingelesenen Zeile kein Contact-Objekt erzeugt werden, so muss eine

FormatException mit entsprechender Message geworfen werden!

- **SaveContactToFile** – speichert das **Contact**-Objekt, das als Parameter mitgegeben wurde in die Datei (**path** entsprechend).

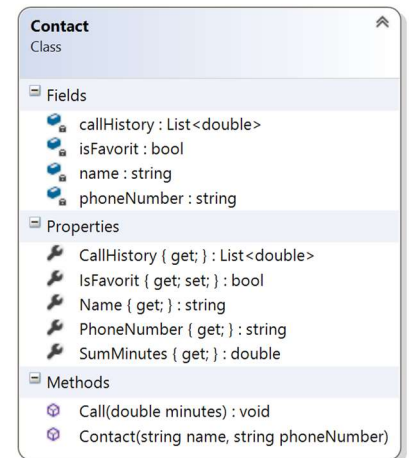
Sollte beim Speichern ein Fehler entstehen, so muss ein eine IOException mit entsprechender Message geworfen werden!

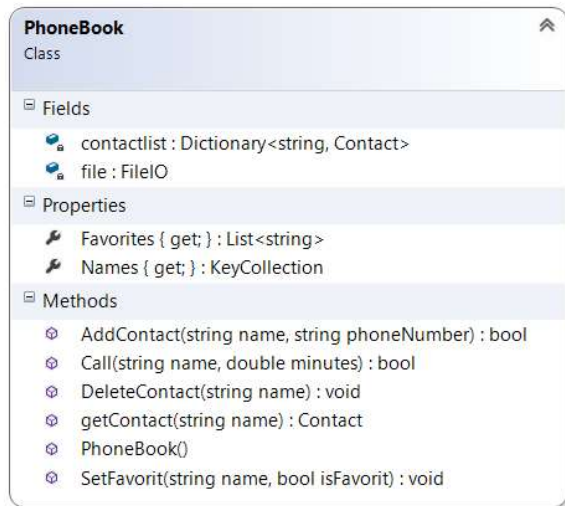
- **RewriteAllContactsToFile** – Speichert alle Contact-Objekt, die als Parameter mitgegeben wurden in die Datei (**path** entsprechend).

Klasse PhoneBook

Die Klasse verwaltet eine Collection von **Contact**-Objekten in einem Dictionary. Key ist der Name des Kontakts. Änderungen an der Collection **contactlist** sollen mit Hilfe der **FileIO**-Klasse auch permanent auf der Festplatte gespeichert werden.

Mit den Objekten aus der Collection (**contactlist**), sind die folgenden Operationen (Methoden und Properties) möglich:





Der *Konstruktor* soll mit Hilfe von *file* die Daten von der Festplatte in die Collection *contactlist* speichern. (Damit wird *contactlist* gleich zu Beginn mit den Daten von der Festplatte befüllt.)

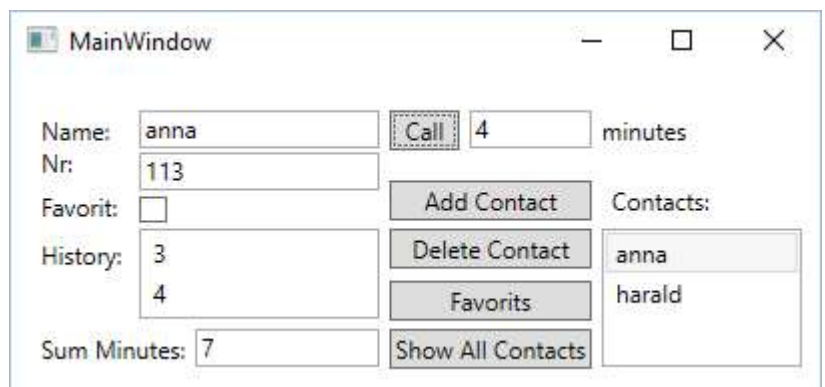
- **AddContact** – fügt den Kontakt dann in die Collection *contactlist* ein, wenn nicht bereits ein anderer Kontakt mit gleichem Namen erfasst wurde. Wenn der Kontakt nicht in die Liste eingefügt wurde, liefert die Methode *false*, ansonsten *true*.
Wenn ein neuer *Contact* in die Collection *contactlist* gespeichert wird, soll dieser *Contact* auch permanent auf der Festplatte gespeichert werden. Verwende dazu die entsprechende Methode von *file*.
- **DeleteContact** – entfernt das *Contact*-Objekt, dessen Name als Parameter mitgegeben wurde, aus der Collection *contactlist*.
- **GetContact** – liefert zu einem Namen das *Contact*-Objekt oder null, wenn kein Objekt gefunden wurde.
- **SetFavorit** – setzt einen Kontakt als Favorit. Dabei gibt *name*, den Namen des Kontakts und *isFavorit*, ob es ein Favorit sein soll oder nicht.

- **Call** – fügt beim Kontakt mit dem Namen *name* in der Anrufliste die mitgegebenen Minuten *minutes* als neuen Eintrag hinzu. Wird unter dem Namen kein Kontakt gefunden, so liefert die Methode *false*, ansonsten *true*.
- **Names** – liefert die *KeyCollection* von *contactlist* (für die Anzeige in der GUI).
- **Favorites** – liefert eine Liste all jener Kontakte, die Favoriten sind (*isFavorit* ist *true*).
- **DeleteContact, SetFavorit, Call** – wenn ein *Contact*-Objekt gelöscht bzw. verändert wird, muss die ganze Datei, in der die Kontaktdaten gespeichert sind, überschrieben werden. Verwende dazu die entsprechende Methode von *file*.

Klasse MainWindow:

In der GUI erfolgt die Interaktion mit dem Benutzer.

- **Add Contact** – es wird ein neues *Contact*-Objekt erfasst und dessen Name in der *ListBox Contacts* angezeigt. Bei fehlerhafter Eingabe bzw. wenn ein neues *Contact*-Objekt nicht eingefügt werden konnte, soll eine entsprechende Meldung angezeigt werden.
- **Delete Contact** – entfernt den selektierten Kontakt on der *ListBox Contacts*, oder liefert eine Fehlermeldung, wenn kein Kontakt ausgewählt wurde.
- Wird ein Kontakt in der *ListBox Contact* selektiert, werden all seine Daten in den Eingabefeldern angezeigt. Dabei wird in der *ListBox History* mit allen Anrufdauern des Kontakts befüllt und ggf. die *Checkbox Favorit* angehakt. Die Anzeige in der *TextBox für Sum Minutes* soll ebenfalls aktualisiert werden.
 - **Favorites** – zeigt nur jene Kontakte an, die Favoriten sind.
- **Show all Contacts** – zeigt alle Kontakte an.
- **Call** – fügt zum selektierten Kontakt die eingegebenen Minuten in der Anrufliste dazu.



Achte darauf, dass das Programm bei keiner Aktion mit einer Exception abstürzen darf sondern dem Benutzer eine aussagekräftige Fehlermeldung anzeigt.

Gutes Gelingen!