

Ein existierendes objektorientiertes System ist zu erweitern.

Beschreibung des schon existierenden Systems

Eine Partnervermittlung hat eine Kartei von alleinstehenden Männern und Frauen (Singles). Jeder Single sucht eine Partnerschaft zu einem anderen Single und hat gegebenenfalls bereits einen Partner gefunden. Ein Single kann

- sich mit einem Angebot auseinandersetzen,
- einem anderen Single einen Antrag machen bzw. einen solchen Antrag von einem anderen Single entgegennehmen und
- sich von einem Partner trennen, um sich mit einem anderen Partner zu verbinden.

Die Partnersuche wird über Botschaften gesteuert.

- Die Vermittlung schickt einem Kunden ein Angebot, das Informationen über einen anderen Single enthält.
- Der Empfänger prüft, ob der angebotene Partner seinen Vorstellungen entspricht und gegebenenfalls seinem derzeitigen Partner vorzuziehen ist. Ist dies der Fall, so macht er dem von der Agentur angebotenen Partner einen Antrag. Der so angesprochene Wunschpartner prüft natürlich auch, ob er sich durch den Antrag verbessern kann. Falls ja, trennt er sich von seinem bisherigen Partner und nimmt den Antrag an. Andernfalls lehnt er den Antrag ab.
- Im Fall eines positiven Bescheides trennt sich dann auch der Antragsteller von seinem bisherigen Partner (sofern er einen hatte), und die neue Partnerschaft ist besiegelt.

Neben Namen und Geschlecht benötigt ein Single weitere Informationen, um sich ein Bild von einem möglichen Partner zu machen. Ein sogenanntes Profil enthält Angaben zu Alter, Größe und Vermögen. Ein Profil kommt aber in zweierlei Bedeutung vor. Zum einen hat jeder Single ein eigenes Profil und zum anderen hat er ein Ideal- oder Wunschprofil seines zukünftigen Partners.

Die Partnerwahl basiert dann im Wesentlichen auf einem Abgleich des Wunschprofils mit dem Eigenprofil eines möglichen Partners.

Abbildung 3.1 (nächste Seite) zeigt das Klassen-Diagramm der Single-Klasse in UML-Notation.

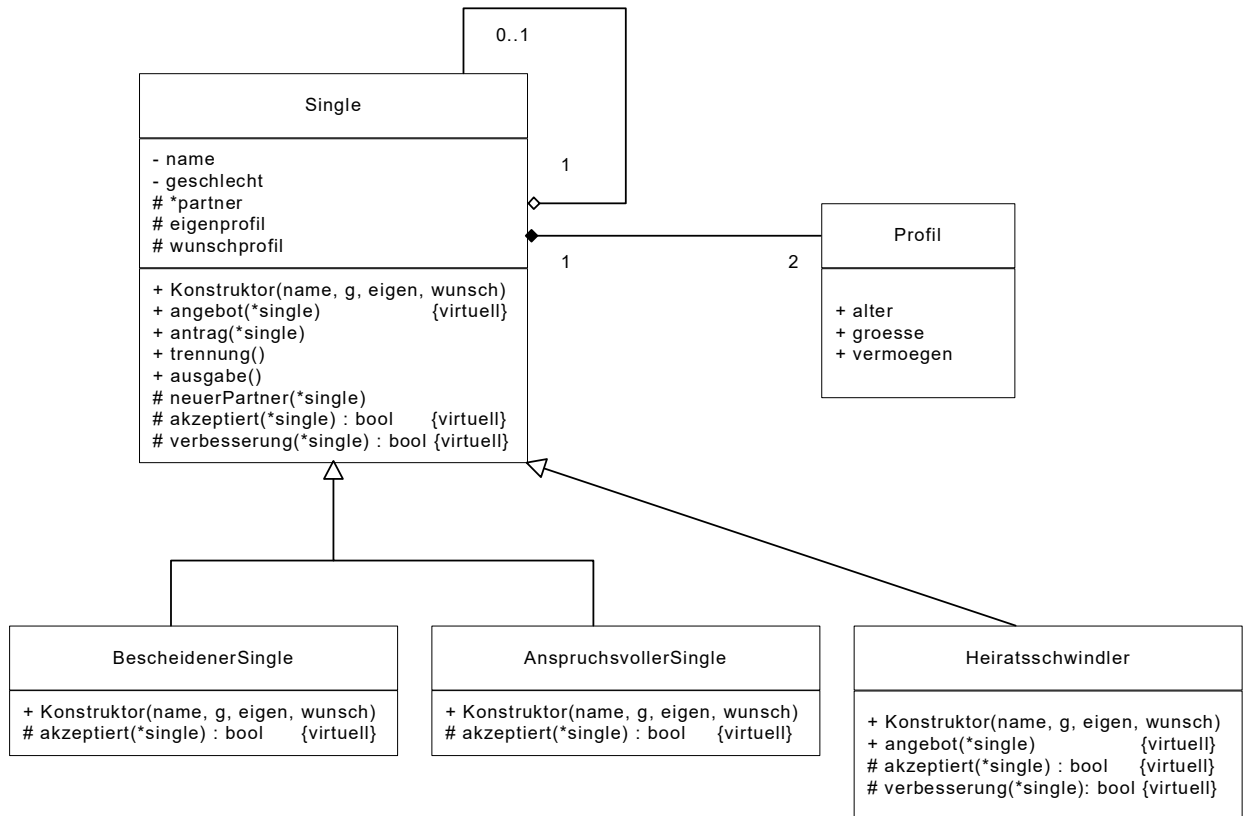


Abb. 3.1 Klassendiagramm des Partnervermittlungssystems.

Ihre Aufgabe

Das System ist um verschiedene Typen von Singles zu erweitern, die unterschiedliche Kriterien bei der Partnersuche anlegen. Zusätzlich zum "gewöhnlichen" Single soll es folgende Spezialfälle geben:

- Der anspruchsvolle Single legt besonders scharfe Kriterien bei der Partnerwahl an.
- Der bescheidene Single ist vollkommen anspruchslos und bereit, jeden als Partner zu akzeptieren.
- Der Heiratsschwindler achtet bei der Partnerwahl nur aufs Geld.

In Abbildung 3.1 sind die Klassenbeziehungen für die neu zu schaffenden Spezialfälle in UML-Notation dargestellt.

Teilaufgabe 3.1

30 Minuten / 15 BE

Analysieren Sie die beigefügten Quelltexte `single.h` und `single.cpp` in Bezug auf die Projektbeschreibung und in Bezug auf Wiederverwendbarkeit und Nachnutzbarkeit von Algorithmen für die nachfolgenden Teilaufgaben.

Legen Sie ein Win32-Konsolen-Projekt `heiraten` mit folgenden Dateien an:

- `single.h`, `single.cpp` (schon vorhanden),
- `sonder.h`, `sonder.cpp` (für die neu zu schaffenden Klassen),
- `heiraten.cpp` (für das Hauptprogramm der Partnervermittlung).

Die Dateien `single.*` sind als abgeschlossen zu betrachten. An ihnen sollen keine Veränderungen vorgenommen werden.

Teilaufgabe 3.2

15 Minuten / 25 BE

Der anspruchsvolle Single verhält sich wie ein normaler Single mit dem einzigen Unterschied: Er akzeptiert Partner nur, wenn die Abweichung des Partnerprofils kleiner als 10 Prozent ist. Er lehnt damit Interessenten ab, die ein gewöhnlicher Single akzeptieren würde (25 Prozent Toleranz).

Definieren Sie in der Datei `sonder.h` eine Klasse `AnspruchsvollerSingle`, der die erforderlichen Methoden umdefiniert. Ein Objekt dieser Klasse erhält bei seiner Instanziierung dieselben Attribute wie ein gewöhnlicher `Single`.

Implementieren Sie die Methoden in der Datei `sonder.cpp`.

Teilaufgabe 3.3

15 Minuten / 15 BE

Der bescheidene Single akzeptiert jedes Angebot, egal welches Profil der Partner hat. Er schaut nicht einmal in das Angebot hinein.

Definieren Sie eine Klasse `BescheidenerSingle` und implementieren Sie ebenfalls die erforderlichen Methoden. Auch diese Objekte übernehmen dieselben Attribute wie Singles.

Teilaufgabe 3.4

30 Minuten / 40 BE

Der Heiratsschwindler hat ganz eigene Vorstellungen über Akzeptanz, Verbesserung und den Umgang mit Angeboten.

Definieren Sie die Klassenschnittstelle des Heiratsschwindlers.

Implementieren Sie alle Methoden des Heiratsschwindlers in der Datei `sonder.cpp`.

Für die von Ihnen durchzuführende Re-Definition der Methoden gelten folgende Vorgaben:

Der Heiratsschwindler akzeptiert nur solche Singles als Partner, deren Vermögen den Wert 50000 (EUR) überschreitet. Alter und Größe sind ihm egal.

Dementsprechend sieht er eine Verbesserung auch ausschließlich darin, einen Partner mit mehr Vermögen zu ergattern. Die Methode `verbesserung(Single* s)` liefert

- `false` bei nicht akzeptierbarem `Single s`,
- `true`, sofern er noch keinen Partner hat, und ansonsten
- `true`, wenn das Vermögen des Singles `s` das Vermögen des bisherigen Partners übertrifft.

Bei einem Angebot zeigt der Heiratsschwindler seine kriminellen Absichten:

Erhält er ein `angebot(Single* s)`, manipuliert er sein eigenes Alter und sein Vermögen entsprechend den Wünschen des Partners, die er ja aus dem Angebot ermitteln kann. Erst danach stellt er dem `Single s` in finsterner Absicht den Antrag. Seine Körpergröße kann er nicht manipulieren, sonst würde er auch das noch tun.

Teilaufgabe 3.5

30 Minuten / 40 BE

Im Hauptprogramm legen Sie zunächst einige `Single`-Instanzen als lokale Variablen mit ihren Profildaten an:

			Eigenprofil			Wunschprofil Partner		
Typ	Name	Geschl.	Alter	Größe	Vermögen	Alter	Größe	Vermögen
Single	Anton	m	55	1.75	100000	50	1.70	0
Single	Berta	w	50	1.70	60000	50	1.80	10000
Heiratsschwindler	Claus	m	30	1.80	100000	25	1.70	0
AnspruchsvollerSingle	Doris	w	60	1.65	100000	65	1.80	10000
BescheidenerSingle	Ernst	m	50	1.80	8000	50	1.80	20000

Tragen Sie die Adressen dieser Singles in je ein Feld aus `Single`-Zeigern für die Männer und für die Frauen ein.

Jetzt kommt der eigentliche Vorgang der Partnervermittlung. Die Agentur macht es sich hier sehr einfach. Sie interessiert sich gar nicht für die Profile der Kunden und wie sie zusammenpassen. Sie bietet der Reihe nach allen Männern alle Frauen an, und danach allen Frauen alle Männer. Diese werden sich dann selbst zusammenfinden.

Implementieren Sie diese Art der Vermittlung.

Rufen Sie im Anschluss für alle Männer und für alle Frauen die Ausgabe-Methode auf, um das Ergebnis des Vermittlungsprozesses auf dem Bildschirm anzuzeigen.

Rufen Sie das Programm `heiraten` in einem Konsolenfenster (MS-DOS-Fenster / Textkonsole) auf. Lenken Sie dessen Bildschirmausgabe in eine Datei `partner.txt` um, ohne am Quelltext des Programmes etwas zu ändern.

Nach: Ulrich Kaiser: C/C++. (200?) S.928ff.