

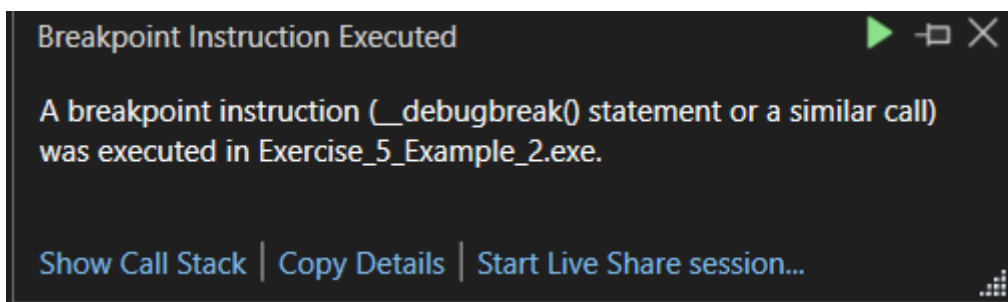
## Inhalt

Beispiel 1) .....	2
Lösungsidee .....	2
Testfälle .....	2
Beispiel 2) .....	6
Lösungsidee .....	6
Testfälle .....	6

### Kommentar:

Ad Beispiel 2)

Da beim delete folgender Fehler aufgetreten ist, wurde zur Freigabe des allokierten Speicherplatzes die erase Operation am Vektor ausgeführt.



Die Fehlermeldung deutet auf einen Zugriff außerhalb der Vector index Range hin (link: <https://stackoverflow.com/questions/72654401/a-breakpoint-instruction-debugbreak-statement-or-a-similar-call-was-execu> ). Nach längerer Analyse/Debugging konnte die Ursache für die Fehlermeldung nicht herausgefunden werden.

Ich würde mich über eine Erklärung freuen, wieso nicht der delete Operator herangezogen werden konnte.

Schließlich sollte sich, laut meinem Verständnis der Index sehr wohl innerhalb des Vectors liegen.

## Beispiel 1)

### Lösungsidee

Über einen Passagier werden Name, Alter, Geschlecht, Kreditkartennummer, Adresse und Informationen über Flug. Die genaueren Informationen über den Flug werden für Hin- und Rückflug als Flugreise gespeichert. Über den Flug werden Flugnummer, Fluggesellschaft, Abflugzeit, Ankunftszeit, Flugdauer und Ziel gespeichert. Bei der Flugreise werden zusätzlich noch die Zwischenziele gespeichert.

Dazu wurde die Komposition verwendet. Eine Komposition ist zu vergleichen mit einer Teil-von-Beziehung. Diese Art wird verwendet, um eine komplexe Instanz zu vereinfachen und aufzuteilen.

### Testfälle

#### 1. Flug mit Hin- und Rückflug

```
Passagier: Anna Anabel<25, weiblich>
Kreditkartennummer: 458652215
Adresse: Straße 10/4/2, 2222 Wien
Hinflug:
Ziel: Las Vegas
Schwechat --> Frankfurt --> Denver --> Las Vegas
Fluggesellschaft: UniFly
Abflugszeit: 14:50
Anflugszeit: 4:50
Flugdauer: 800
Flugnummer: 4563
Rückflug: Las Vegas --> San Francisco --> München --> Linz
Fluggesellschaft: FlyHigh
Abflugszeit: 8:10
Anflugszeit: 21:40
Flugdauer: 810
Flugnummer: 8598
```

#### 2. Kein Rückflug

```
Microsoft Visual Studio Debug Console

Passagier: Anna Anabel<25, weiblich>
Kreditkartennummer: 458652215
Adresse: Straße 10/4/2, 2222 Wien
Hinflug:
Ziel: Las Vegas
Schwechat --> Frankfurt --> Denver --> Las Vegas -->
Fluggesellschaft: UniFly
Abflugszeit: 14:50
Anflugszeit: 4:50
Flugdauer: 800
Flugnummer: 4563
Achtung: Rückflug noch nicht gebucht!
```

### 3. Keine Informationen zum Passagier

```
Passagier: Not given Not given<Not given, Not given>
Kreditkartennummer: 0
Adresse: Not given
Hinflug:
Ziel: Las Vegas
Schwechat --> Frankfurt --> Denver --> Las Vegas
Fluggesellschaft: UniFly
Abflugszeit: 14:50
Anflugszeit: 4:50
Flugdauer: 800
Flugnummer: 4563
Rückflug: Las Vegas --> San Francisco --> München --> Linz
Fluggesellschaft: FlyHigh
Abflugszeit: 8:10
Anflugszeit: 21:40
Flugdauer: 810
Flugnummer: 8598
```

### 4. Direktflug

```
Passagier: Anna Anabel<25, weiblich>
Kreditkartennummer: 458652215
Adresse: Straße 10/4/2, 2222 Wien
Hinflug:
Ziel: Las Vegas

Fluggesellschaft: UniFly
Abflugszeit: 14:50
Anflugszeit: 4:50
Flugdauer: 800
Flugnummer: 4563
Rückflug:
Ziel: Schwechat

Fluggesellschaft: FlyHigh
Abflugszeit: 8:10
Anflugszeit: 21:40
Flugdauer: 810
Flugnummer: 8598
```

## 5. Alter fehlt

```
Passagier: Anna Bell<Not given, weiblich>
Kreditkartennummer: 45632
Adresse: Strasse ,2222 Wien
Hinflug:
Ziel: Las Vegas
Schwechat --> Frankfurt --> Denver --> Las Vegas
Fluggesellschaft: UniFly
Abflugszeit: 14:50
Anflugszeit: 4:50
Flugdauer: 800
Flugnummer: 4563
Rückflug:
Ziel: Schwechat
Las Vegas --> San Francisco --> München --> Linz
Fluggesellschaft: FlyHigh
Abflugszeit: 8:10
Anflugszeit: 21:40
Flugdauer: 810
Flugnummer: 8598
```

## 6. Geschlecht fehlt

```
Passagier: Anna Bell<25, Not given>
Kreditkartennummer: 45632
Adresse: Strasse ,2222 Wien
Hinflug:
Ziel: Las Vegas
Schwechat --> Frankfurt --> Denver --> Las Vegas
Fluggesellschaft: UniFly
Abflugszeit: 14:50
Anflugszeit: 4:50
Flugdauer: 800
Flugnummer: 4563
Rückflug:
Ziel: Schwechat
Las Vegas --> San Francisco --> München --> Linz
Fluggesellschaft: FlyHigh
Abflugszeit: 8:10
Anflugszeit: 21:40
Flugdauer: 810
Flugnummer: 8598
```

## 7. Kein Name

```
C| Passagier: Not Given<25, weiblich>
| Kreditkartennummer: 44586589
| Adresse: Strasse 10, 2222 Wien
| Hinflug:
| Ziel: Las Vegas
| Schwechat --> Frankfurt --> Denver --> Las Vegas
| Fluggesellschaft: UniFly
| Abflugszeit: 14:50
| Anflugszeit: 4:50
| Flugdauer: 800
1| Flugnummer: 4563
1| Rückflug:
1| Ziel: Schwechat
1| Las Vegas --> San Francisco --> M³nchen --> Linz
| Fluggesellschaft: FlyHigh
| Abflugszeit: 8:10
| Anflugszeit: 21:40
| Flugdauer: 810
| Flugnummer: 8598
```

## Beispiel 2)

### Lösungsidee

Bei einer Vererbung werden Komponenten einer Klasse von einer anderen Klasse übernommen also vererbt. Es werden Bereiche (private, protected, public) aus der Eltern-Klasse vererbt. Die wird beim Deklarieren der Kind-Klasse angemerkt welcher Bereich von einer oder mehreren Klassen vererbt wird.

Funktionen mit demselben Namen und derselben Schnittstelle können in der Kind-Klasse überschrieben werden, dazu wird das Schlüssel „override“ verwendet. Wenn „override“ verwendet wird, muss in der Eltern-Klasse „virtual“ verwendet werden. Die Funktion wird als Virtualfunktion bezeichnet. Eine Pure Virtualfunktion wird = 0 gesetzt. Wenn eine Basisklasse Virtualfunktionen besitzt wird diese auch als abstrakte Klasse benannt.

### Testfälle

#### 1. Alle Parts eines Tisches

```
Microsoft Visual Studio Debug Console
Tisch: Tischplatte, Bein (gross), Bein (gross), Bein (gross), Bein (gross)
```

#### 2. In eine Datei speichern

- Hierarchische Ausgabe

	A
	Tischplatte
	Bein (gross)
	Bein (gross)
	Bein (gross)
	Bein (gross)

- Set Ausgabe

	A	
1	1 Tischplatte	
2	4 Bein (gross)	

#### 3. Aus einer Datei lesen

```
Microsoft Visual Studio Debug Console
Tischplatte; Bein (gross); Bein (gross); Bein (gross); Bein (gross); (gross);
```

#### 4. Hierarchische ausgegeben

- Auf Konsole

```
Tischplatte; Bein (gross); Bein (gross); Bein (gross); Bein (gross);
```

- In Datei

	A
1	Tischplatte
2	Bein (gross)
3	Bein (gross)
4	Bein (gross)
5	Bein (gross)

## 5. Set Ausgabe

- Auf Konsole

```
1 Tischplatte;
4 Bein (gross);
```

- In Datei

	A
1	1 Tischplatte
2	4 Bein (gross)

## 6. Part-Funktionen

- Copy Konstruktor

```
part is a copy of part
```

- Equal (positives Beispiel)

```
part Equals part
```

- Equal (negatives Beispiel)

```
part Does Not Equal equipment
```

- GetName

```
The Name of the Part is part
```

## 7. Composite Part -Funktionen

- Konstruktor mit Part

```
Konstruktor mit String: Part
```

- Konstruktor mit Vektor

```
Konstruktor mit Vektor: new part new part new part
```

- Neue Parts hinzufügen

```
New Part added: Part addedPart
```

- GetParts

```
GetParts: new part new part new part
```