

+... public  
 -... private  
 #protected

```
Public class Shape{
    Public void draw(){
        System.out.print("s");
    }
}
```

```
Public class Triangle extends shapes
    @override
    Public void draw(){
        Sysout("Tri");
    }
}
```

Was hat das s für einen statischen Datentyp? Shape

```
Shape s = new Shape();
s.draw()
Triangle t = new Triangle();
t.draw()
s = t;
t = s; -> nicht möglich
```

s.draw(); -> triangel wird ausgegeben weil dynamisch gebunden

Hat hier unten immernoch den Datentyp shape

```
s = new Rectangle();
s.draw();
```

Polymorphie ändert seine Form

s=t; -> geht  
 s.bimbam(); -> Compiler weiß es nicht und kann daher nur shape Methoden aufrufen  
 t.bimbam(); -> funktioniert

Was heißt dynamisch?

Zur kompilzeit?

Beim nasenbohren?

Zur laufzeit?

Was heißt statisch?

Zur kompilzeit

Normalerweise macht man Attribute private

Wenn Vererbung dann protected damit die Unterklassen darauf zugreifen können

### 1. Simulationsumgebung (6 Punkte)

Entwickeln Sie eine Klasse Trolley, die folgenden Operationen zur Verfügung stellt:

- Bewegen innerhalb des Verteilzentrums (Das Verteilzentrum ist als rechteckiges Netz von immer gleich großen Zellen aufgebaut)

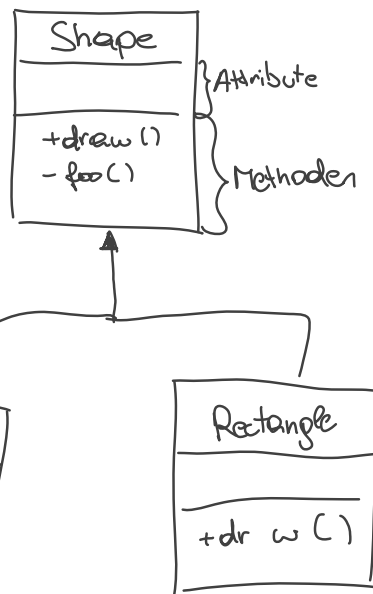


- Laden eines Produktes
  - Das erst geladene Produkt definiert das Ziel, das der Trolley ansteuert. Es dürfen nur Produkte dazu geladen werden, die das gleiche Ziel haben.
- Entladen aller Produkte: Am Bildschirm soll eine Liste der ausgeladenen Produkte ausgegeben werden.

Überlegen Sie sich für Ihr Design das Konzept eines „Transportauftrags“. Auch die Definition eines sinnvollen Aufbaus eines Verteilzentrums ist Teil dieser Aufgabe!



Trolley sollte Position wissen  
 Kirsche hat ein Ziel  
 muss nicht selbstständig fahren können  
 man muss aber sagen können fahre nach Norden



abstract

```
Public abstract class Shape{
    Public abstract void draw(){
    }
}
```

```
Public class Triangle extends shapes
    @override
    Public void draw(){
        Sysout("Tri");
    }
}
```

würde beim UML Diagramm kursiv sein

Shape

## 2. Ein Fall fürs Standesamt (7 Punkte)

Sie werden vom Gemeindeamt Ihres Heimatortes beauftragt, das Standesamt bei der Erledigung von Heirats- und Scheidungsangelegenheiten softwaremäßig zu unterstützen:

Für alle *Personen* werden im Standesamt folgende Daten gehalten:

- Vor- und Familienname (bei verheirateten Frauen auch deren Mädchennamen),
- Geschlecht
- Geburtstag
- Bei verheirateten Personen auch ein Verweis auf den jeweiligen Partner.

*Heiraten* ist in Österreich nur zwei verschiedengeschlechtlichen, vorher unverheirateten Partnern erlaubt, die mindestens 18 Jahre alt sind. Bei Jüngeren wäre eine Ehe mit einer speziellen Zustimmungserklärung möglich – aber wer will denn das überhaupt?

Bei der Heirat müssen auch die Trauzeugen vermerkt werden. Dabei gilt, dass auch diese mindestens 18 Jahre alt sein müssen.

Eine *Scheidung* zweier verheirateter Partner ist jederzeit möglich und bringt beide zurück in den Zustand, den sie vor der Ehe hatten – zumindest vor dem Standesamt ;-). Wesentlich bei der Scheidung ist auch die Angabe des Scheidungsgrundes.

Modellieren Sie Personen, indem Sie eine Klasse `Person` so konstruieren, dass Sie diese auf folgende Art und Weise benutzen können:

```
Person donald = new Person("Donald", "Duck", male, "1.1.1980");
Person daisy  = new Person("Daisy", "Queen", female, "10.1.1982");
...
donald.marry(daisy); // has same effect as: daisy.marry(donald);
```

Elementare Algorithmen und Objektorientierte Programmierung

1/2

Datentyp Date  
Klasse Kalender  
oder selber implementieren

Konstruktor Geburtsdatum und das ist als String  
java Kalender!!!! Google suche

daisy sollte ja auch verheiratet mit donald sein sowie umgekehrt  
donald.marry(daisy)

```
void marry(Person p){ -> p sollte daisy sein
    _partner = p;
    p._partner = this;
```

(geht auch mit set partner)

```
donald.divorce("Vernachlässigung"); // has same effect as:
daisy.divorce("Vernachlässigung");
```

Natürlich müssen alle Fehlerfälle abgefangen werden!

## 3. Vererbung - Bibliothek (7 Punkte)

In einer Bibliothek wird eine Reihe von verschiedenen Leihobjekte angeboten. Darunter fallen: Bücher, Zeitschriften, Videos, CD-ROMs und Videospiele (Sega, Playstation, Nintendo, Xbox, ...). Zu jedem Leihobjekt sind abhängig von Typ verschiedene Informationen wie z.B. ISBN, Verlag, ... für Bücher, Interpret, Musikrichtung, ... bei Musik CDs zu speichern. Jedes Leihobjekt besitzt aber auf jeden Fall eine eindeutige Nummer. Zusätzlich sollen auch Personen und deren Leihen erfasst werden können.

Entwickeln Sie eine Klassenhierarchie für die Leihobjekte eines Bibliotheksystems in Form eines UML-Klassendiagramms. Überlegen Sie sich die sinnvolle Verwendung von Vererbung! Denken Sie auch an die Erweiterbarkeit des Angebots der Bibliothek.

diese eindeutige nummer statisch machen (static number) counter immer um 1 erhöhen

statisch unterstreichen  
unterstrichenes Attribut = statisch