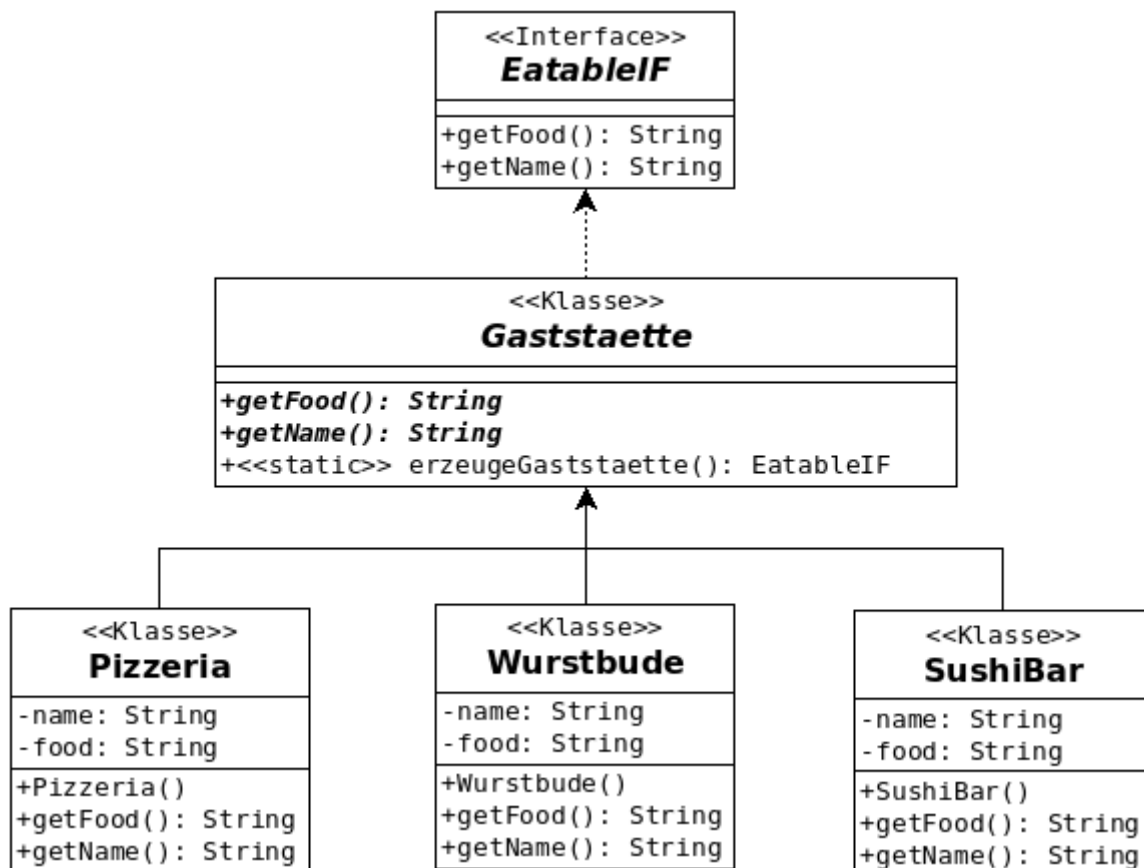


1. Klassendiagramm



2. Namen für die Gaststätten

```
private String name;
private String[] namen = {"DA MICHELE", "CONCETTINA AI TRE SANTI", "NAPOLI",
                          "LA GROTTA", "VALENTINA", "LA CAMPANGOLA"};

//Konstruktor
public Pizzeria()
{
    name=this.getClass().getSimpleName()+" "+namen[r.nextInt(namen.length)];
    food= ...
}
```

3. Erzeugung der Gaststätten

Factory-Methode

Entwurfsmuster, bei dem durch Aufruf einer statischen Methode der Oberklasse Objekte der abgeleiteten Klassen instanziiert werden

Ziel: zentralisierte Erzeugung von Objekten

kein direkter Aufruf des Konstruktors - Subklassen müssen nicht bekannt sein

einfache Wartung des Quellcodes, z.B. beim Hinzufügen neuer Klassen

```
public static EatableIF erzeugeGaststaette()  
{  
    //erzeuge Zufallszahl n  
    ...  
    switch(n)  
    {  
        case 0:      e=new Wurstbude();  
                     break;  
        case 1:      e=new Pizzeria();  
                     break;  
        ...  
    }  
    return e;  
}
```

4. Gaststättenliste

Erzeugung einer zufälligen Anzahl von Gaststätten mit Hilfe der Factory-Methode

Speicherung in einer Liste

```
//Gaststaetten-Liste  
List<EatableIF> lokale = new ArrayList<>();  
  
//zufaellige Anzahl Gaststaetten erzeugen und speichern  
for(int i=0;i<anzahl;i++)  
    lokale.add(Gaststaette.erzeugeGaststaette());
```

Ausgabe der Liste:

```
//mit For-Each-Schleife  
for(EatableIF e: lokale)  
{  
    System.out.println(e.getName());  
    System.out.println(e.getFood());  
};  
  
//mit Iterator  
Iterator<EatableIF> it = lokale.iterator();  
while (it.hasNext())  
{  
    EatableIF e=it.next();  
    System.out.println(e.getName());  
    System.out.println(e.getFood());  
}  
  
//mit Zählschleife (wenn der Index von Bedeutung ist)  
for (int i=0;i<lokale.size();i++)  
{  
    System.out.println(lokale.get(i).getName());  
    System.out.println(lokale.get(i).getFood());  
}
```

5. HashMap

Verwaltung von Key-Value-Paaren

jeder Schlüssel existiert in der HashMap nur einmal

beim Einfügen eines Key-Value-Paars, dessen Schlüssel in der HashMap bereits existiert, wird der Wert überschrieben

Methoden

```
public V put(K key, V value)           //speichert Key-Value-Paar; falls key
                                       //schon vorhanden, wird value ersetzt
                                       //und der alte Wert zurueckgegeben,
                                       //andernfalls null

public V get(Object key)               //liefert value zu übergebenem key
                                       //(oder null)

public boolean containsKey(Object key) //ermittelt, ob key bereits enthalten ist

public Set<K> keySet()                 //liefert Menge der Schluessel
```

6. Speicherung der Gaststätten nach Strassen

HashMap mit **Key:** Strasse
 Value: Gaststättenliste

```
Map< String, List<EatableIF> > stadt = new HashMap<>();

do
{
    //...Strasse eingeben

    if (!stadt.containsKey(strasse))
    {
        //erzeuge Gaststaetten-Liste lokale fuer die Strasse
        ...
        stadt.put(strasse, lokale);
    }
    else
        System.out.println("Strasse bereits vorhanden!");

    ...
}while (wdh=='J');
```

7. Ausgabe der HashMap

Ausgabe der Gaststätten mit Name und Speisekarte

Über eine HashMap kann man nicht direkt iterieren.

Man kann sich aber mit der Methode keySet() die Schlüsselmenge beschaffen und diese durchgehen:

```
//Ausgabe der HashMap mit Iteratoren
Iterator<String> itKeys= stadt.keySet().iterator();
while(itKeys.hasNext())
{
    //Strasse
    String str;
    System.out.print("\nStrasse:  ");
    str=itKeys.next();
    System.out.println(str);
}
```

```

//Gaststaetten-Liste
System.out.print("Gaststaetten:\n ");
Iterator<EatableIF> itValue = stadt.get(str).iterator();
while(itValue .hasNext())
{
    EatableIF e = itValue.next();
    System.out.println(e.getName());
    System.out.println(e.getFood());
}
}
}

```

alternativ:

```

//Ausgabe der HashMap mit For-Each
for(String str : stadt.keySet())
{
    System.out.println("\nStrasse: "+str);
    System.out.println("Gaststaetten:");
    for(EatableIF e: stadt.get(str))
    {
        System.out.println(e.getName());
        System.out.println(e.getFood());
    }
}
}

```