

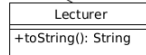
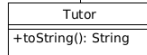
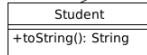
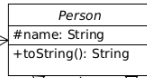
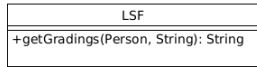
# Observer-Pattern

---

Carsten Gips (HSBI)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

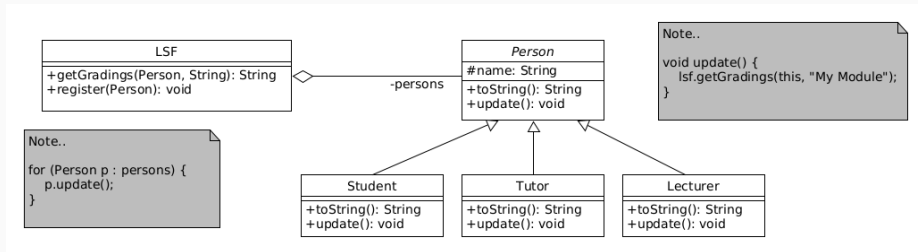
# Verteilung der Prüfungsergebnisse



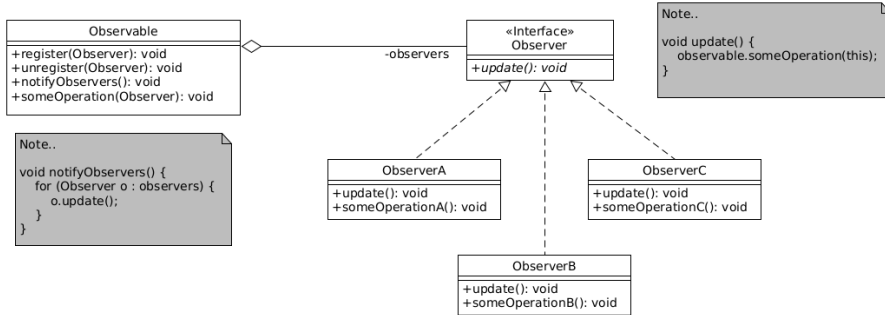
Note..

```
for (Person p : persons) {
    lsf.getGradings(p, "My Module"); // ???!
}
```

# Elegantere Lösung: Observer-Entwurfsmuster



# Observer-Pattern verallgemeinert



Observer-Pattern: Benachrichtige registrierte Objekte über Statusänderungen

- Interface `Observer` mit Methode `update()`
- Interessierte Objekte
  1. implementieren das Interface `Observer`
  2. registrieren sich beim zu beobachtenden Objekt (`Observable`)
- Beobachtetes Objekt ruft auf allen registrierten Objekten `update()` auf
- `update()` kann auch Parameter haben

# LICENSE



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.