

# Calendar Project

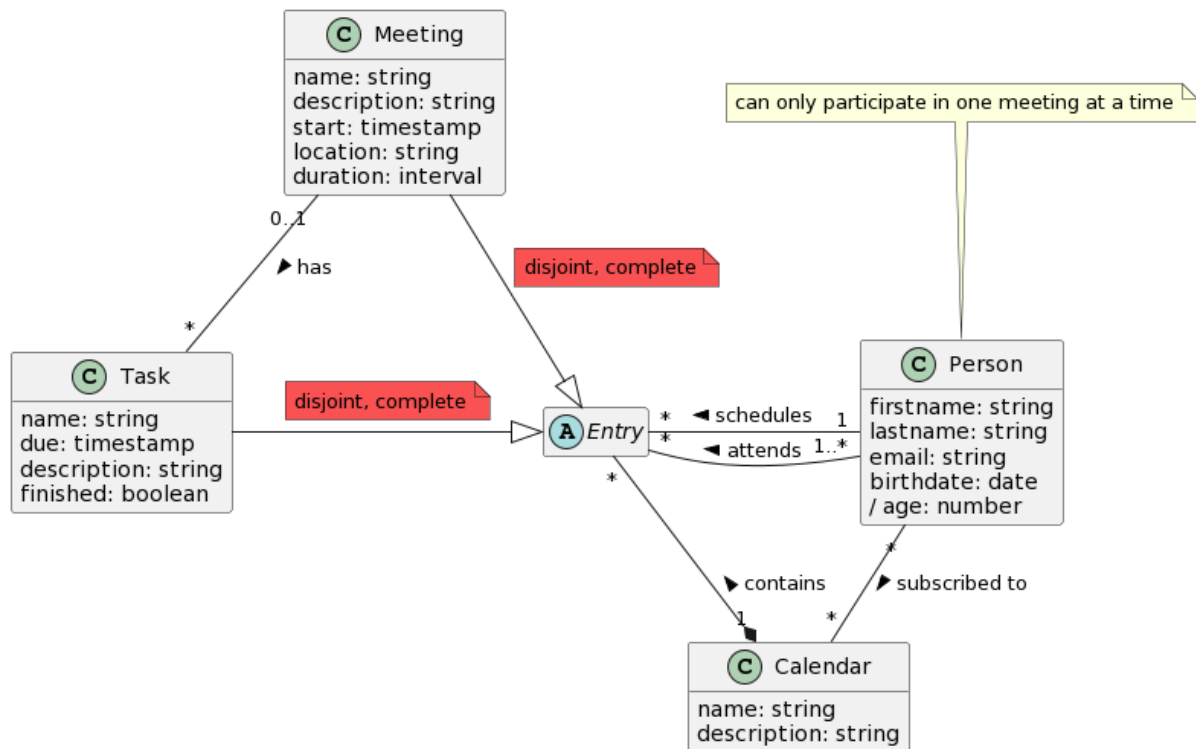
Oliver Dietsche, Simon Peier

## Prosa

Dieses System ermöglicht das Verwalten von Aktivitäten in einem Kalender.

- Meetings erhalten eine eindeutige ID. Es kann also mehrere Meetings mit dem gleichen Namen geben.
- Das Alter einer Person ist dynamisch und wird anhand des Geburtstages berechnet und nicht direkt in der Datenbank gespeichert.
- Meetings haben neben dem Organisator mindestens einen weiteren Teilnehmer.
- Eine Person kann zu jedem Zeitpunkt nur an einem Meeting teilnehmen.
- Wenn ein Task erfüllt ist, wird dieser nicht gelöscht, sondern mit Hilfe eines Flags als "finished" markiert.

## Domain Model



## Relationales Modell

```
Person (  
    ID INTEGER,  
    Firstname TEXT NOT NULL,  
    Lastname TEXT NOT NULL,  
    Email TEXT NOT NULL UNIQUE,  
    Birthdate DATE  
);
```

```
Person_Calendar (  
    ID INTEGER ,  
    Person INTEGER NOT NULL,  
    Calendar INTEGER NOT NULL  
);
```

```
Calendar (  
    ID INTEGER,  
    Name TEXT NOT NULL,  
    Description TEXT NOT NULL  
);
```

```
Person_Meeting (  
    ID INTEGER,  
    Person INTEGER NOT NULL,  
    Meeting INTEGER NOT NULL  
)
```

```
Meeting (  
    ID INTEGER,  
    Name TEXT NOT NULL,  
    Start TIMESTAMP NOT NULL,  
    Duration INTERVAL NOT NULL,  
    Description TEXT,  
    Location TEXT,  
    Calendar INTEGER NOT NULL,  
    Scheduler INTEGER NOT NULL  
)
```

```
Task (  
    ID INTEGER,  
    Name TEXT NOT NULL,  
    Due TIMESTAMP NOT NULL,  
    Finished BOOLEAN NOT NULL,  
    Description TEXT,  
    Calendar INTEGER NOT NULL,  
    Assignee INTEGER NOT NULL,  
    Meeting INTEGER  
)
```

### Begründung relationale Abbildung der UML-Vererbung

Wir haben uns bei der Abbildung unserer Vererbung für Regel 3.b / Variante 2 entschieden und somit pro Subklasse je eine Tabelle erstellt (Meeting, Task). Auf die Abbildung der Superklasse haben wir verzichtet, da die beiden Subklassen nur 3 von jeweils 7 Attributen teilen.