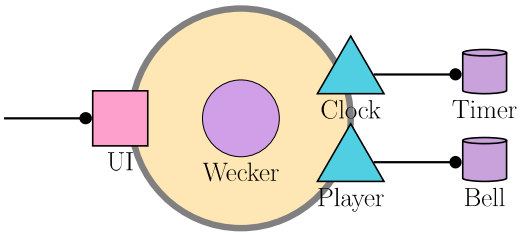
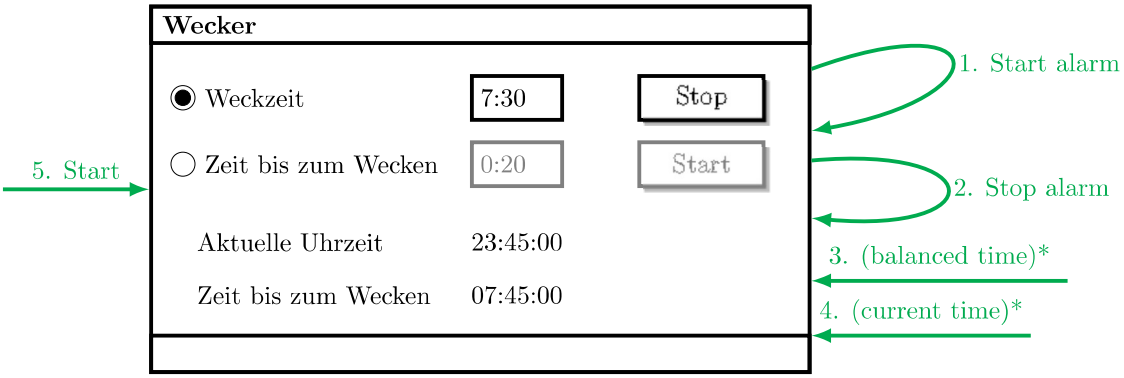
1.4 Der Wecker

1.4.1 Analyse

System:



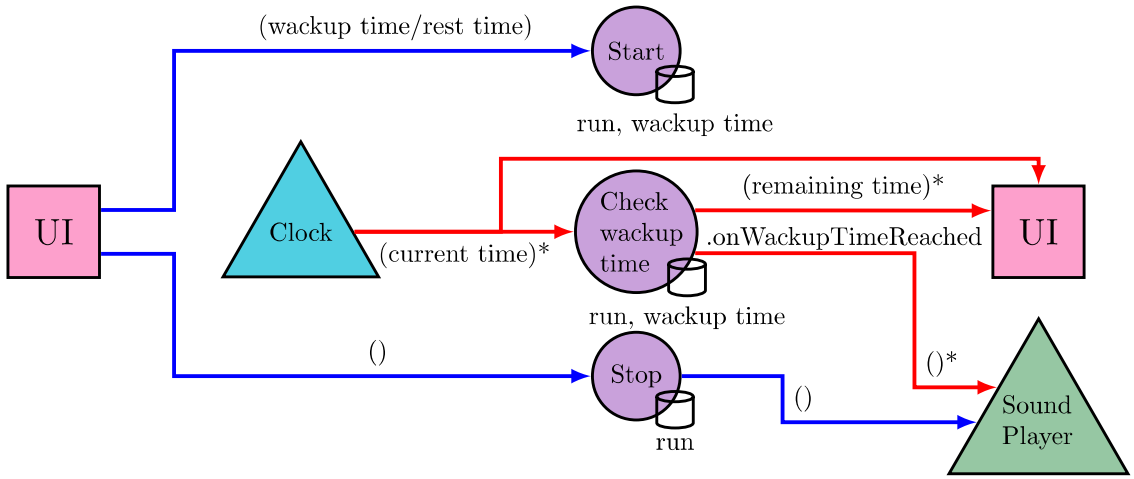
Anwendung: X
Co-Worker/Module: X
Dialoge: Wecker (UI)
Interaktionen:



Nachrichten:

Alter Zustand	Eingabe	Nachricht	Restzeit	Neuer Zustand
1. Wecker aus, Zeit 22 Uhr	7:30 Uhr	Start Alarm mit Weckzeit(WZ)	9h 30min	Wecker läuft
1. Wecker aus, Zeit 22 Uhr	2h	Start Alarm mit Ruhezeit(RZ)	2h	Wecker läuft
2. Wecker läuft, Zeit 0 Uhr, WZ 7:30 Uhr	-	Stop Alarm	-	Wecker aus
2. Wecker läuft, Zeit 7:30 Uhr, WZ 7:30 Uhr	-	-	0s	Ton spielt, Wecker läuft
2. Wecker läuft, Zeit 7:30 Uhr, WZ 7:30 Uhr	-	Stop Alarm	-	Ton aus, Wecker aus
5. Wecker aus, Zeit 22 Uhr	-	-	-	Wecker aus

1.4.2 Design



Schnittstellen:

```
struct Watchdog {
  void StartWithWakupTime(DateTime);
  void StartWithRestTime(TimeSpan);
  void Stop();
  void CheckWackupTime(DateTime);
  Event<> onWackupTimeReached;
  Event<TimeSpan> onRemainingTime;
};
```

```
struct UI {
  void UpdateTime(DateTime);
  void UpdateRemainingTime(TimeSpan);
  void Start();
  Event<DateTime> onStartWithWackupTime;
  Event<TimeSpan> onStartWithRestTime;
  Event<> onStopped;
};
```

```
struct Clock {
  void Start();
  void Stop();
  Event<DateTime> onCurrentTime;
};
```

```
struct SoundPlayer {
  void Play();
  void Stop();
};
```

1.4.3 Hinweise

- Das ist eine Übung, die die letzten Schritte der Analyse, das Design und die Integration des Systems (Programmaufbau, Nachrichten, Logik) üben soll.
- Fokus liegt hier auf den Nachrichtenaustausch, da eher weniger Logik vorhanden ist.
- Das erste Inkrement sollte nur sein, die Aktuelle Uhrzeit anzuzeigen. Die Implementierung der Uhr und die Verknüpfung mit der UI im richtigen Moment hat seine Tücken.
- Empfehlenswert für Azubis mit UI-Framework Erfahrung
- Achte auf Verwendung der IODA Architektur
- Erst wenn das Design und die Schnittstellen fertig sind, sollte man die Azubis anfangen lassen das Projekt aufzusetzen.
- Mit den Akzeptanztests beginnen lassen