Задание 6

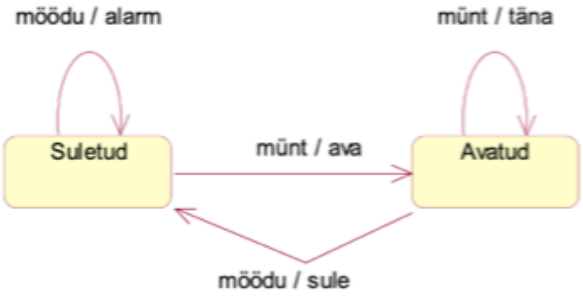
State и Observer Oleku GoF шаблоны

- Представить дизайн (class & interaction diagram) и реализацию примера с турникетом, используя State GoF pattern. Методы класса «pöördvärav (турникет)»): ava (открыть), sule (закрыть), täna (поблагодарить) и alarm (тревога) должны просто отображать на экране соответствующие действия.

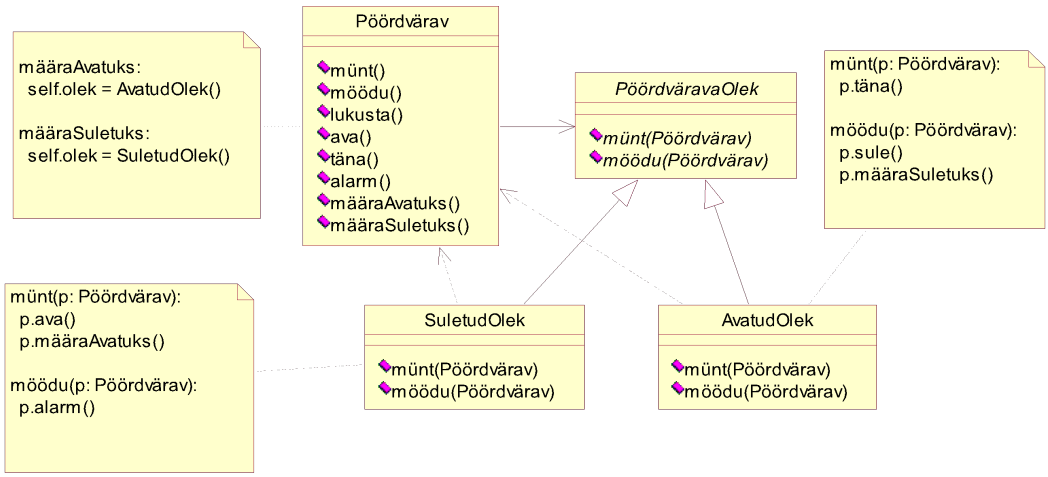
- Добавить дизайн (class & interaction diagram) и реализацию к предыдушему примеру турникета, который позволяет добавлять к турникету разных наблюдателей, которые извещаются о тревоге. Необходимо использовать GoF observer pattern, где наблюдаются турникет. Наблюдатель должен отображать уведомление на экране.

ПРИМЕР

- Имеем дело с программой платного турникета:.
    a. Состояния: avatud (открыто), suletud (закрыто)
    b. Регистрируемые события: münt (монета), möödu (проходи)
    c. Действия: ava (открыть), sule (закрыть), täna (поблагодарить плательщика), alarm (тревога).
- Как избежать условной логики в классе pöördvärav(турникет)?



ПЕРЕВОД С ЭСТОНСКОГО:
Möödu/alarm – проходи/тревога
Münt/täna – монета/поблагодарить
Münt/ava – монета/открыть
Möödu/sule – проходи/закрыть
Avatud – открыто
Suletud – закрыто



НИЖЕ ПЕРЕВОД ДИАГРАММ

Pöördvärav(турникет)
münt() – монета
möödu() – проходи
lukusta() – захлопнуть/закрыть на замок
ava – открыть
täna – поблагодарить
alarm() – тревога
sule() – закрыть
määraAvatuks() - назначить открытым
määraSuletuks() - назначить закрытым

määraAvatuks	- назначитьОткрытым
self.olek=AvatudOlek()	olek – состояние, avatud - закрыто
määraSuletuks	- назначитьЗакрытым
self.olek=SuletudOlek()	olek – состояние, avatud - закрыто