

# Обектно-ориентиран анализ и проектиране на софтуерни системи с UML

Зимен семестър 2022/2023

проф. д-р Боян Бончев ([bbontchev@fmi.uni-sofia.bg](mailto:bbontchev@fmi.uni-sofia.bg)) и  
ас. д-р Явор Данков ([yavor.dankov@fmi.uni-sofia.bg](mailto:yavor.dankov@fmi.uni-sofia.bg))

Всяко упражнение от тази дисциплина се състои от три части. Някои от тях са за групово решаване. За да Ви бъдат най-полезни те, е необходимо да се разделите на групи от 3-5 човека. Първата част е за групово дискусия, втората е за самостоятелно усвояване на инструменти и техники, а третата е за работа в екип и развитие на уменията за анализ и проектиране.

Материалите за този курс са достъпни от  
<https://learn.fmi.uni-sofia.bg/course/view.php?id=8633>

## Упражнение Модул 7: Нововъведения в UML 2.0 – времеви диаграми и език за ограничения на обектите (OCL)

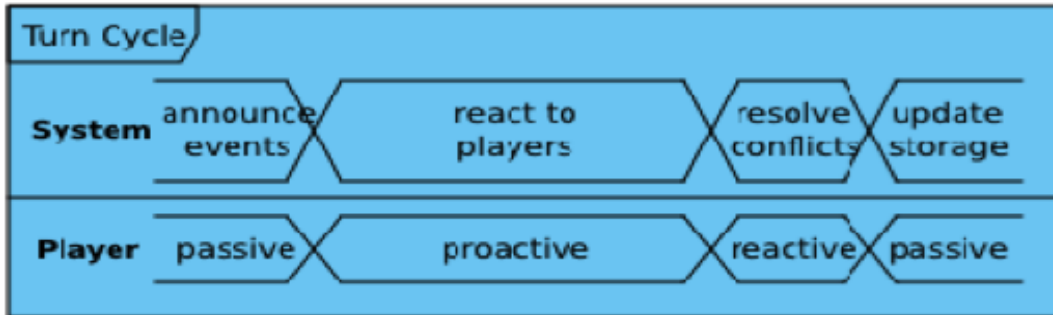
### Съдържание

Времеви диаграми.

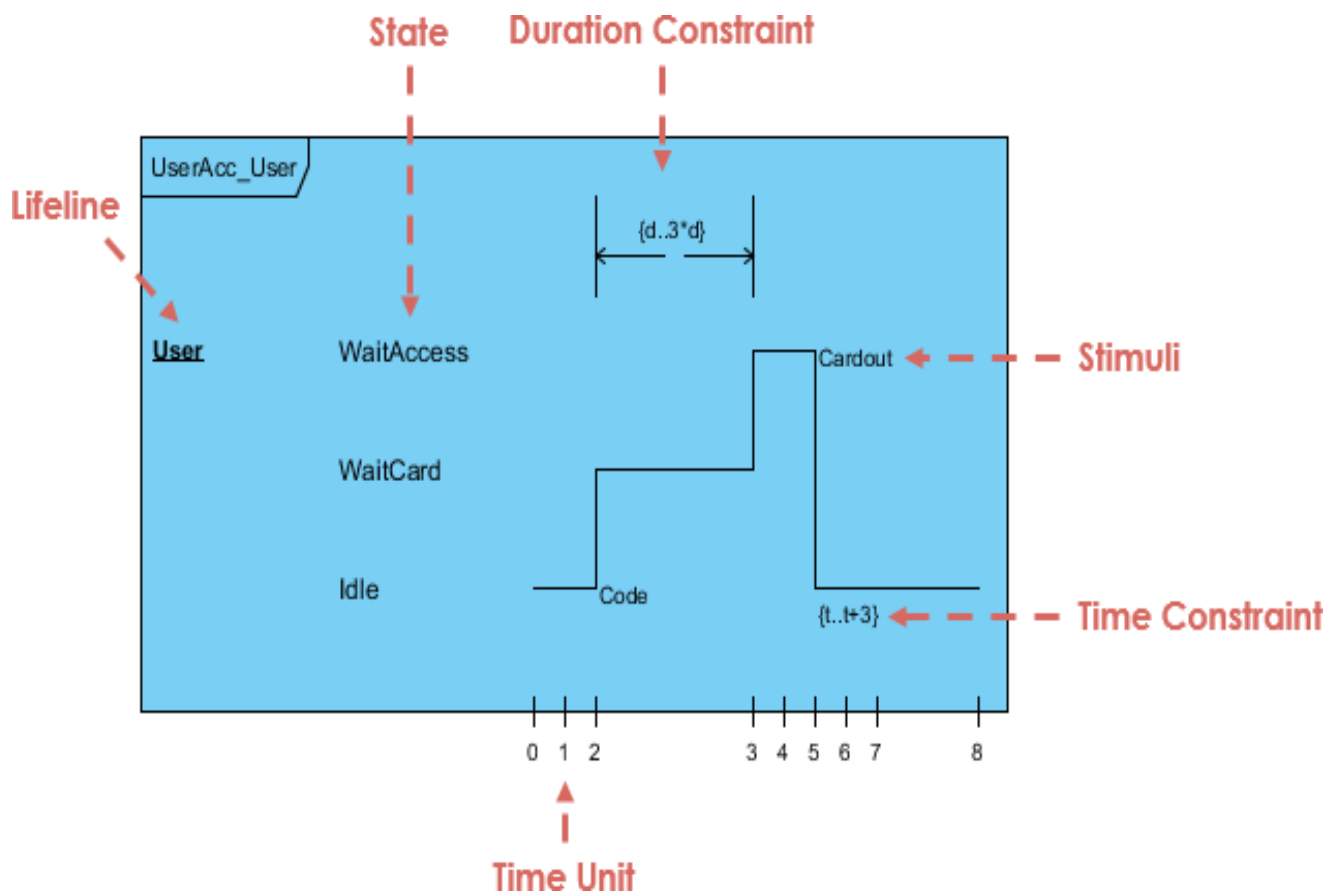
Език за описване на ограничения на обектите.

### Част А: Групово дискусия

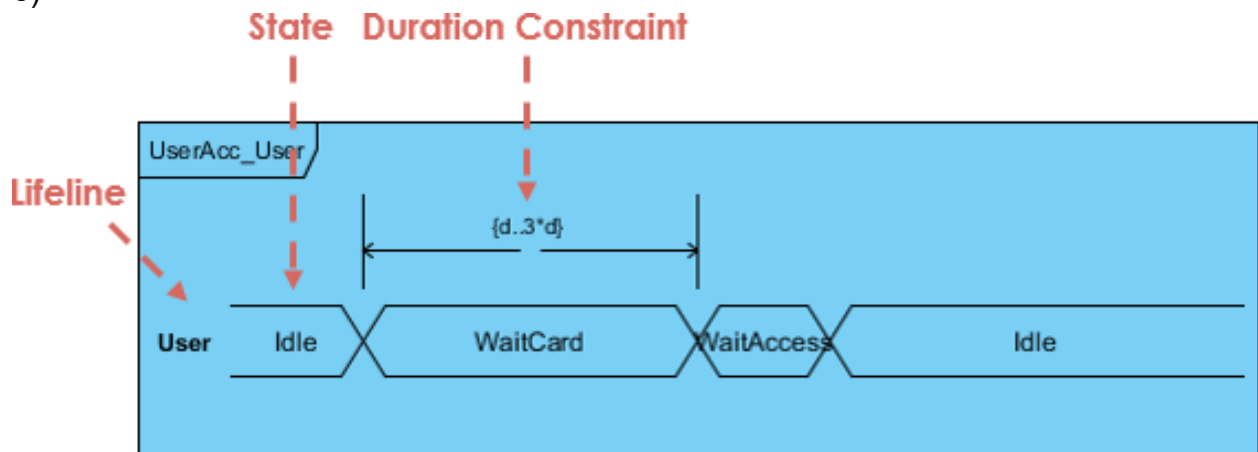
1. Времевият аспект на какъв процес описва дадената диаграма? Синхронна или асинхронна е комуникацията между играча и системата? Защо смятате така? Има ли възможност същата диаграма да бъде използвана за другия вид комуникация?



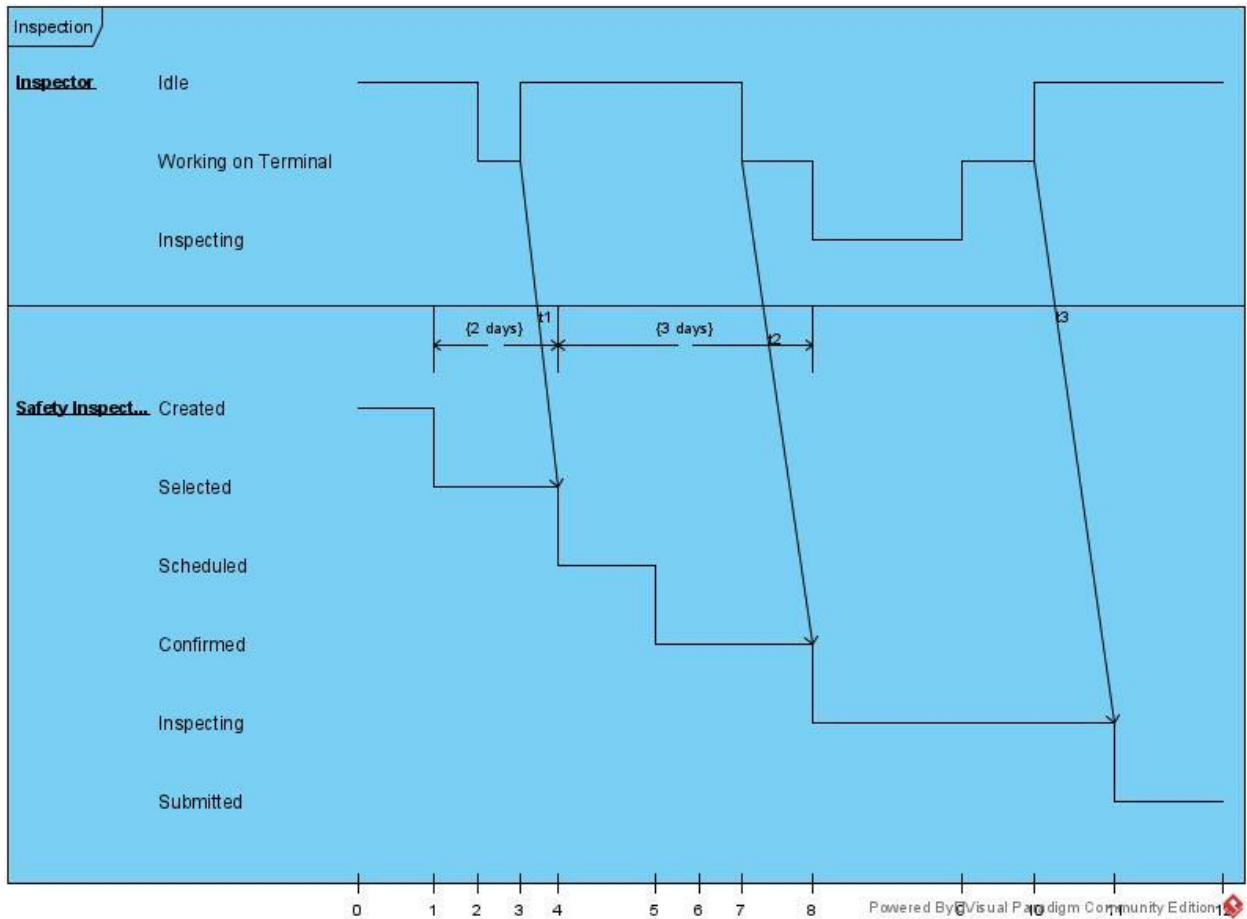
2. Можете ли да формулирате с думи каква е разликата между разширената и компактната форма на времевите диаграми? Кой елемент засяга разликата? Има ли обстоятелства, при които разширената форма носи повече информация, отколкото компактната и каква би била тази информация?
3. Коментирайте следните примерни диаграми (Visual Paradigm), тяхното значение и характеристики?
  - a)



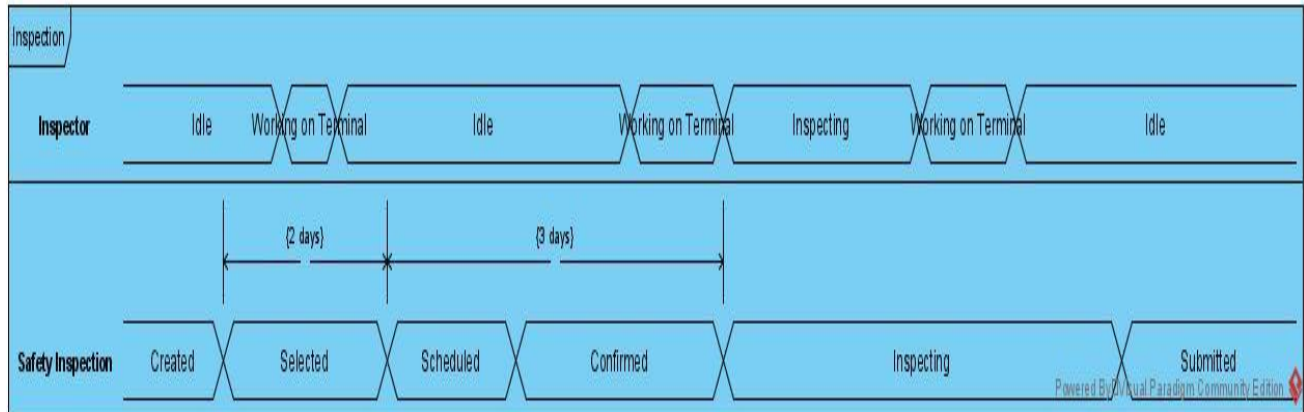
6)



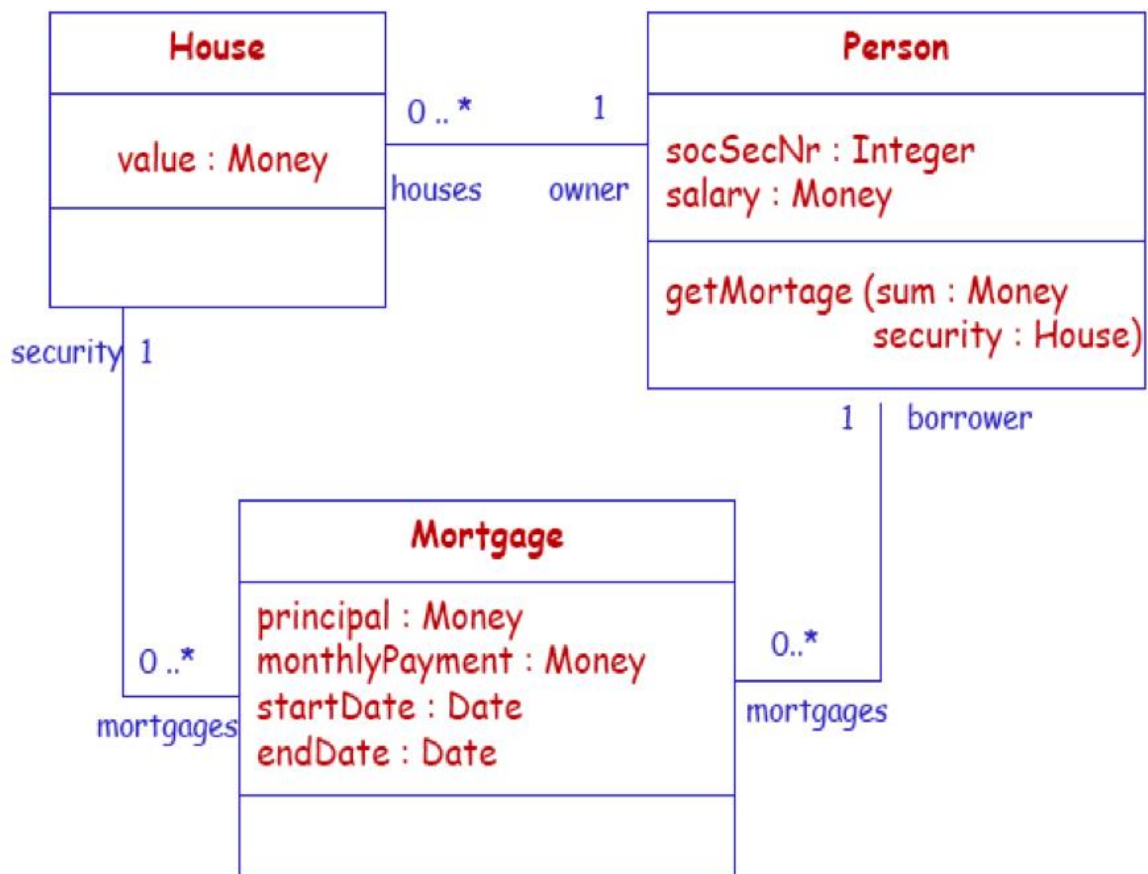
B)



г)



4. Разгледайте диаграмата и ограниченията. Формулирайте всяко от тях като изречение в свободен текст, като се стараете да запазите семантиката.



```
context Mortgage
inv: security.owner = borrower
```

```
context Mortgage
inv: startDate < endDate
```

```
context Person
inv: Person::allInstances()->isUnique(socSecNr)
```

```
context Person::getMortgage(sum : Money, security : House)
pre: self.mortgages.monthlyPayment->sum() <= self.salary * 0.30
```

```
context Person::getMortgage(sum : Money, security : House)
pre: security.value >= security.mortgages.principal->sum()
```

<http://www.klasse.nl/ocl/ocl-reasons.html>

Допълнителна информация за времеви диаграми и OCL има в глава 9 и приложение A в книгата Learning UML 2.0 на Хамилтън и Майлс.

## Част Б: Инструменти и техники

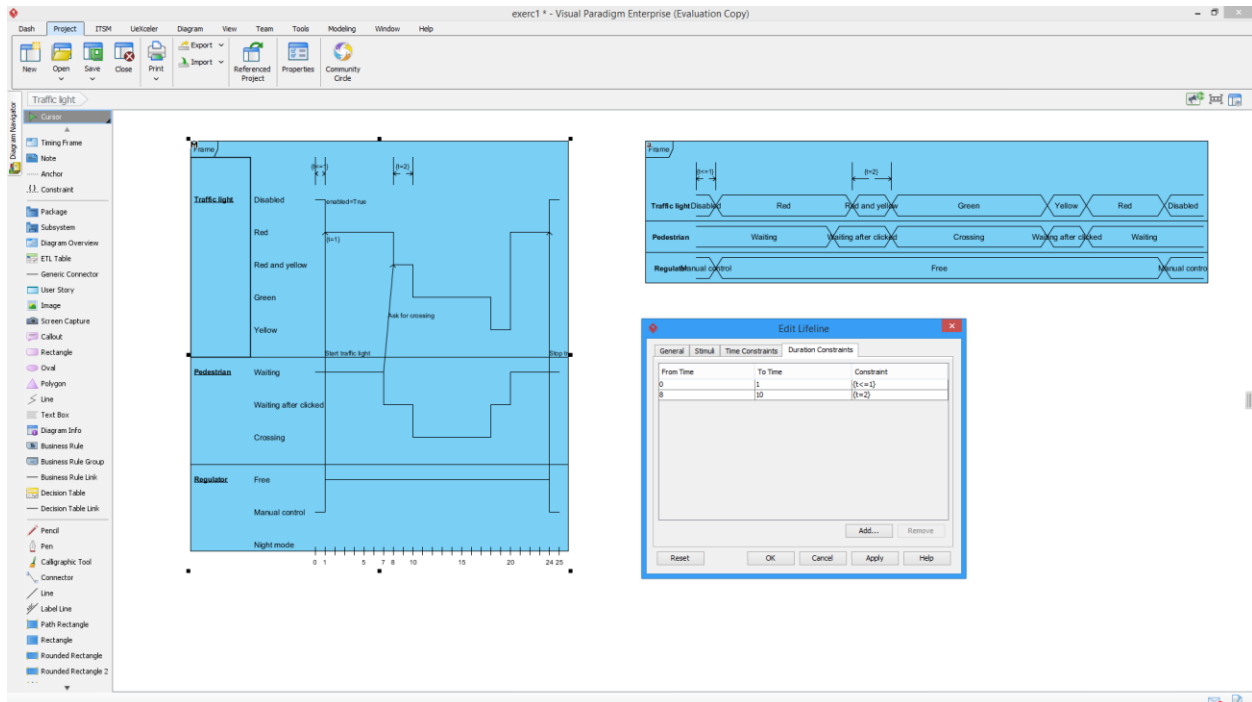
Създайте времева диаграма на светофар като използвате диаграмата на машината на състоянията.



Преминаването между състоянията трябва да става за по малко от 1сек. В зелено състояние светофара трябва да се намира 25сек, в червено 55. Опитайте следното:

- Добавете един пешеходен светофар с бутон за искане на зелена светлина за пешеходците
- Добавете регулировчик
- Изпробвайте добавяне на съобщения (събития) към диаграмата
- Добавете времеви ограничения (timed constraints)
- Добавете ограничения за продължителност (duration constraints)
- Копирайте диаграмата и представете копието в компактен вид. Анализирайте разликите между двете представяния.

Фигурата по-долу представя примерни времеви диаграми, описващи горните изисквания.

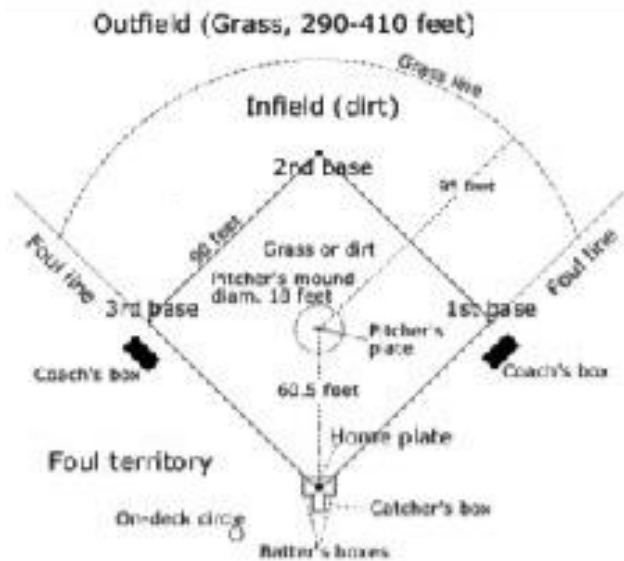


При необходимост от помощ вижте упътванията на адрес [https://www.visual-paradigm.com/support/documents/vpuserguide/94/2586/6715\\_drawingtimin.html](https://www.visual-paradigm.com/support/documents/vpuserguide/94/2586/6715_drawingtimin.html)

## Част В: Анализ и проектиране

Запознайте се с правилата на бейзбола.

Целта в бейзбола е да се отбележат повече точки. Точка се взема, когато играчът направи пълна обиколка /рън/ на четирите бази, които лежат по краищата на поле с формата на квадрат. Играе се от два отбора, всеки от по 9 играча. Единият отбор подава топката и защитава полето, а другият - удря топката и се стреми да завземе повече бази и да отбележи точка. Всеки мач се състои от 9 ининга, като при всеки ининг ролите на нападащ и защитаващ се отбор се сменят.



Играта започва когато играчът от защитаващия полето отбор /питчерът/, стоящ в центъра на бейзболното поле, се опитва да подаде топката на своя съотборник /кетчер/ през играч от противниковия отбор, заел позиция пред него на хоума от отбора на съперника и държащ бата /батер/. Целта на кетчера е да защитава хоума като лови топките, които батерът изпуска. Останалите играчи от защитаващия отбор нямат точно определени позиции, но не може да са между питчера и батера.

Ако батерът удари топката, побягва, за да завземе първа база. Ако негови съотборници вече са завзели бази /рънъри/, те също могат да тръгнат към следваща база, за да не бъдат изместени когато батерът стигне до тях. Защитаващият полето отбор се стреми да улови ударената топка и бързо да я подаде към защитника на базата, към която батерът или друг негов съотборник тича. Ако топката бъде уловена във въздуха, батерът изгаря. Ето защо, се избягват удари с голяма парабола, които дават възможност това да стане. Когато играч на нападащия отбор бъде докоснат от съперника с топката, той е изгорен. Поради това, бягащите до базите понякога ги завземат с пързаляне /слайдинг/- за да не бъдат докоснати с топката.

Ако батерът не удари топката при 3 "добри" хвърляния (страйка -en. strike), е вън от играта и на негово място застава следващият батер. Трябва да се знае, че батер не е точно определена длъжност. Така се нарича всеки противников играч, заел позиция на началната точка /хоума/. При трима играчи в аут, инингът свършва и започва друг, като ролите на двата отбора се сменят. За "добро" се счита това хвърляне на питчера, което попада в определена зона, ограничена между коленете и гърдите на батера, над хоума. Ако питчер допусне 4 "лоши" ("бол" - en.ball) хвърляния, батерът получава служебно първа база. Играчите по базите се наричат съответно първи бейзмен, втори бейзмен и трети бейзмен.

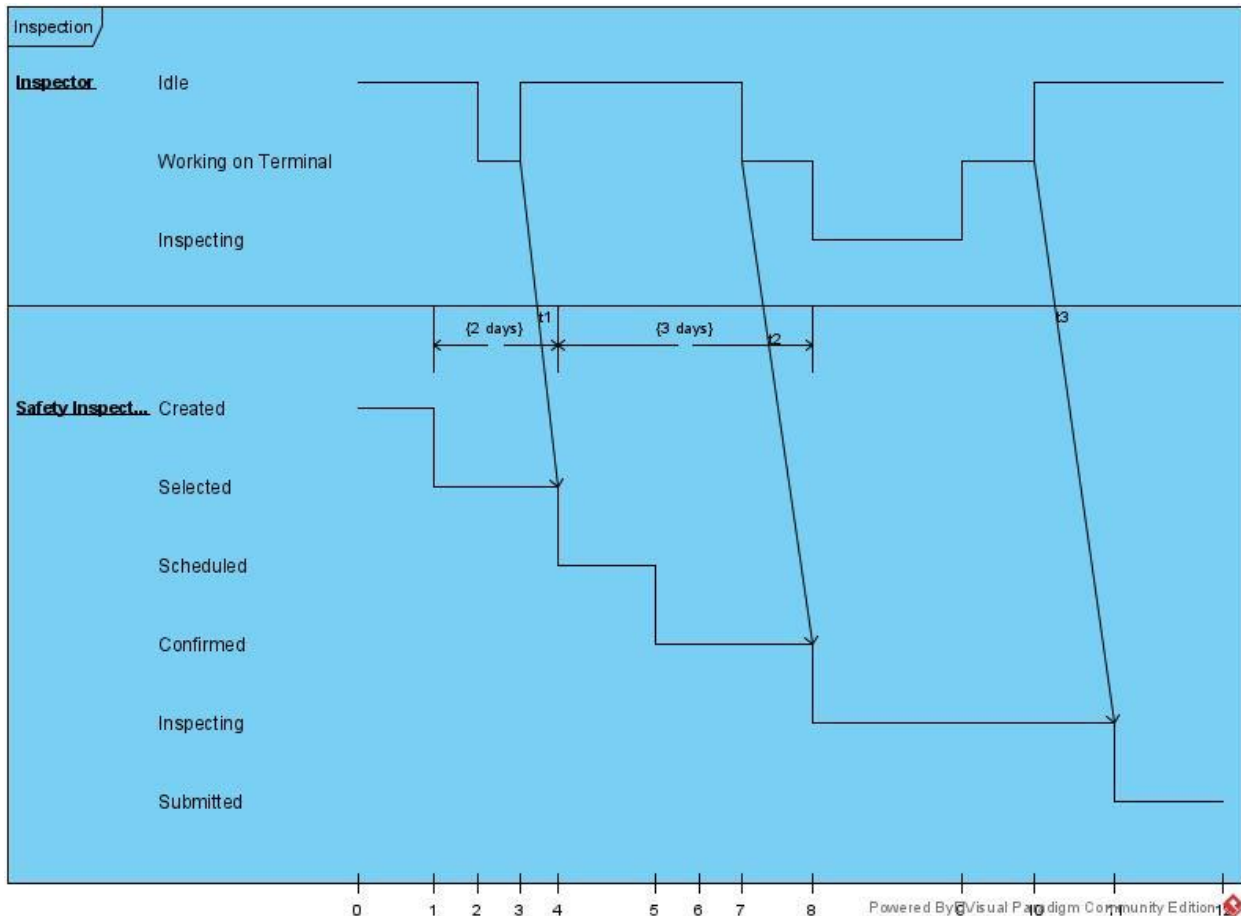
Ако батерът обиколи всички бази и се върне в хоума за един удар, той регистрира хоум рън.

Направете времева диаграма за случая на хоум рън. Можете ли да дадете примери за инвариант, предусловие и след-условие в тази игра и да ги формулирате в синтаксиса на OCL?

### Част Г: Анализ на примерни диаграми

Прегледайте и анализирайте, представените времеви диаграми от Част Б. Опитайте се да отговорите на поставените въпроси като съставите кратък текст, който дава описание на състоянията, през които преминава системата. Документирайте преходите между състоянията и тяхната семантика, отчитайки времевите метрики, представи в диаграмата. В диаграмата използван ли е езикът за ограничения на обектите (OCL) и каква е семантиката му в съответната диаграма? Документирайте описанието на системата. Има ли възможности за подобряване на диаграмите?

#### Времева диаграма 1:





## Времева диаграма 2:

