

Диаграма на машина на
състояние

State Machine diagram

State machines diagrams

- ❑ State machines diagrams показват промените в състоянието на обекта и събитията, които са предизвикали тези промени
 - ❑ State machine diagrams се използват в UML, за да моделират динамичното поведение на един обект по време на няколко потребителски случая
-

State

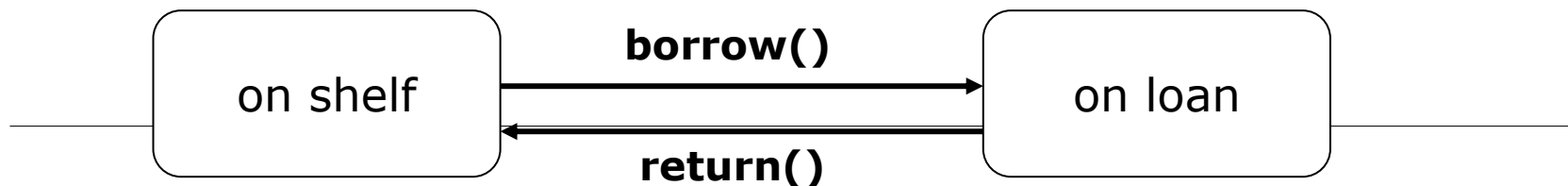
- *"State е състояние или ситуация по време на живота на даден обект, по време на които той удовлетворява някакви условия, извършва някаква дейност или изчаква за някакво събитие"*

[Booch , "UML Language User Guide"]



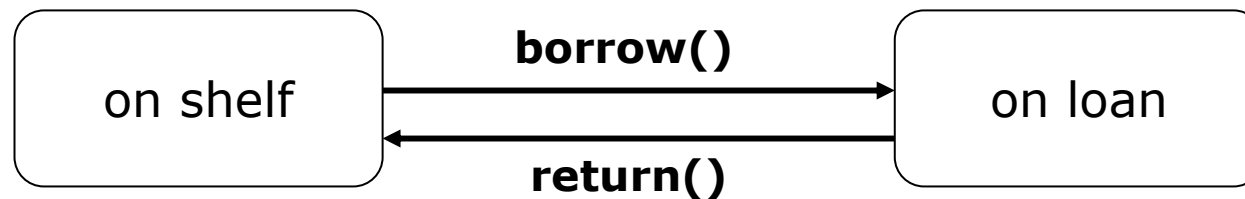
Преход между състоянията

- **Transition** (преход) - връзка между две състояния, показваща, че обектът в първото състояние ще извърши някакво действие и ще премине във второто състояние, когато определено множество от събития и условия са удовлетворени.
 - преход между състоянията в отговор на някакво събитие
 - Изобразява се като плътна линия със стрелка в единия край, сочеща следващото състояние.
- Пример - класът **Copy** представлява единично копие на книга
 - състояния: 'onShelf' и 'onLoan'

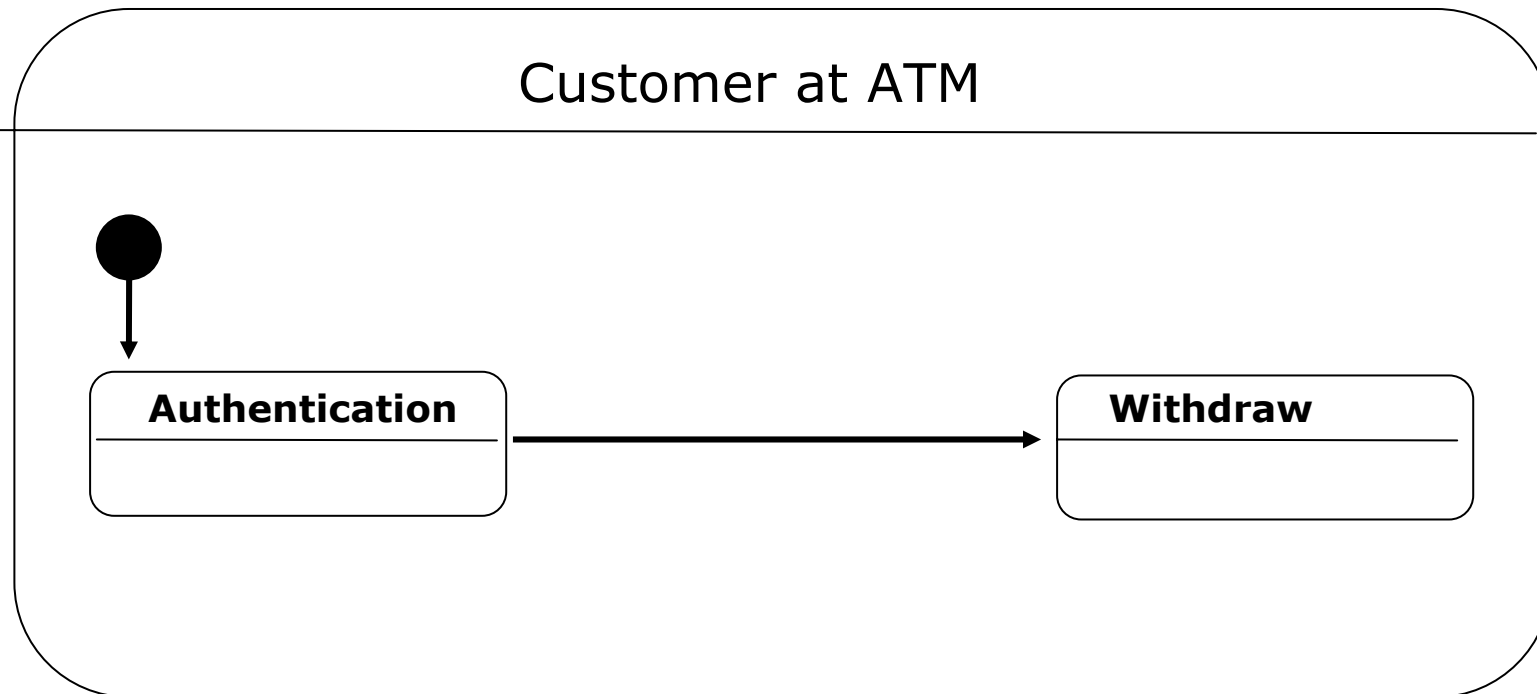


Събитие

- **Event** - важно събитие, което задейства прехода между състоянията.

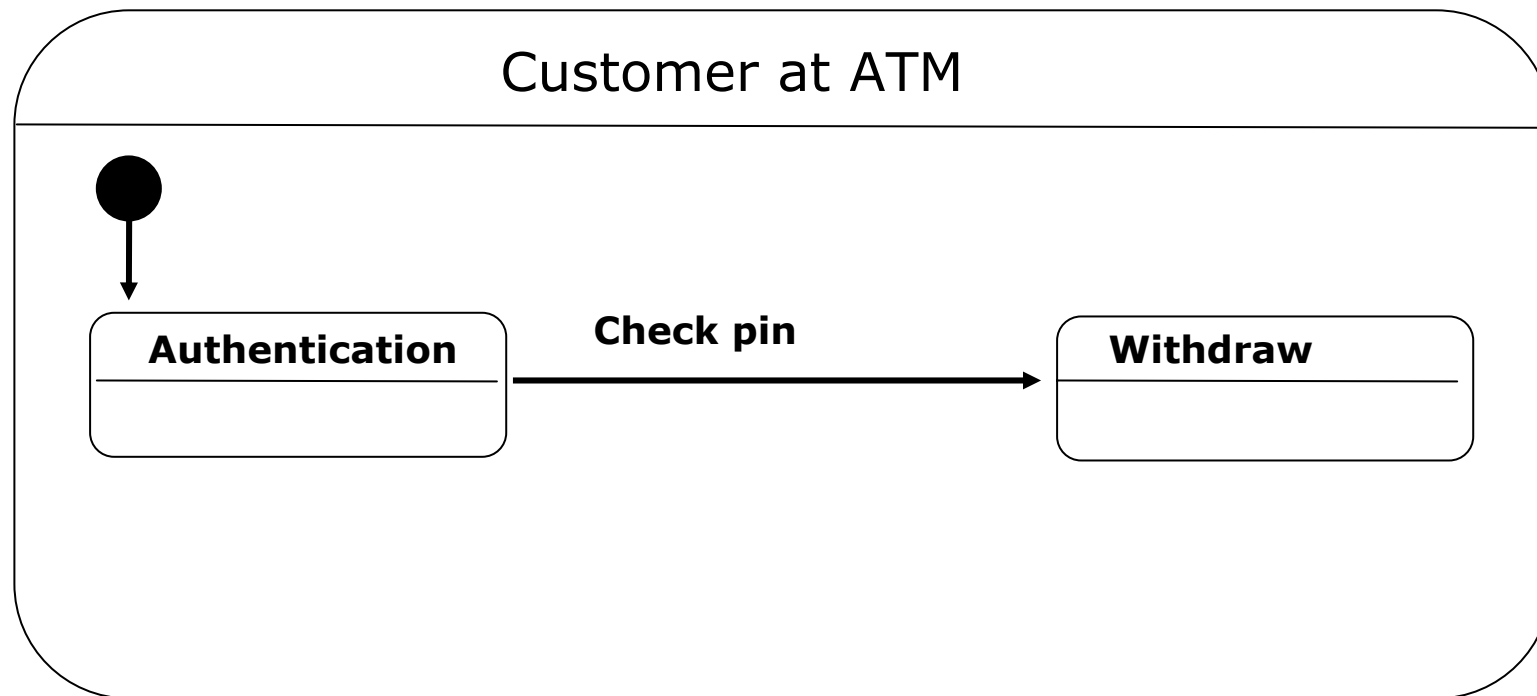


ATM Example



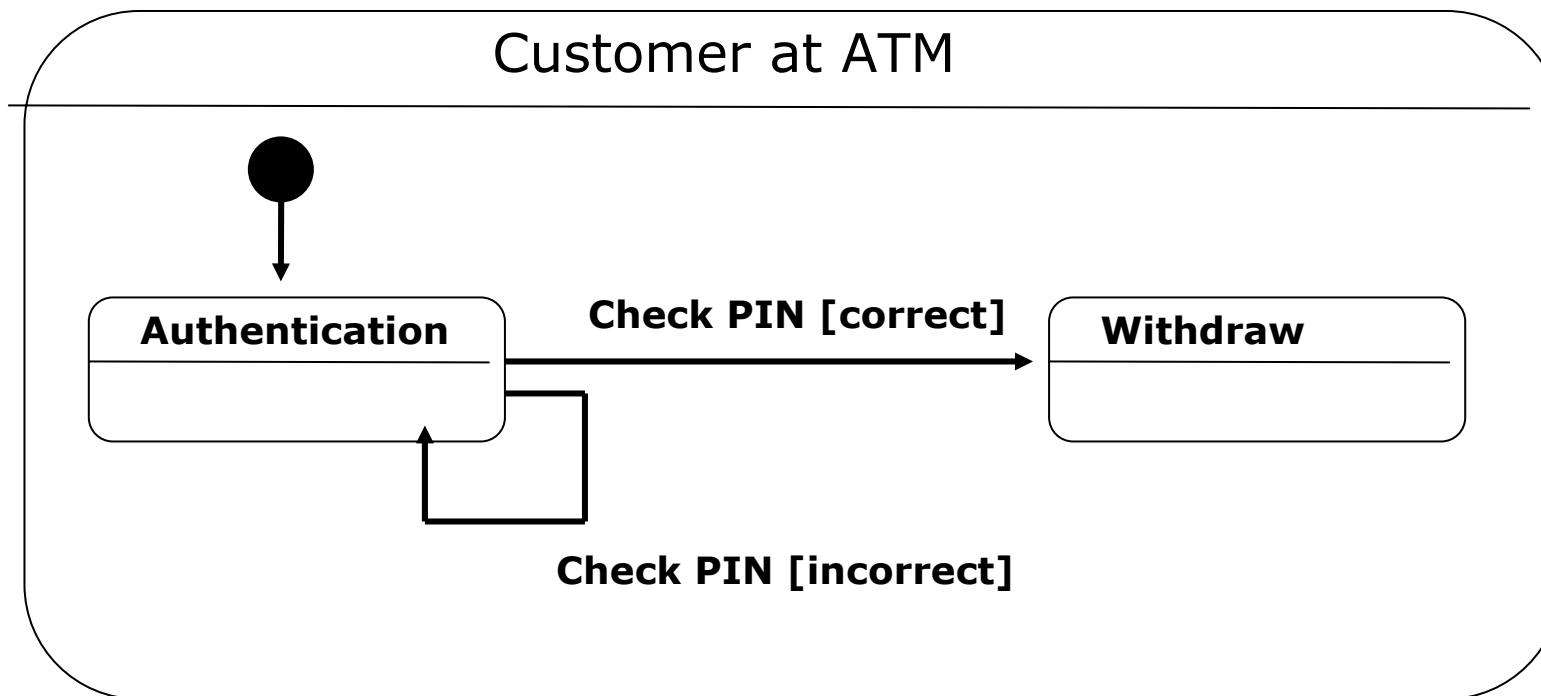
- ❑ Клиентът трябва да премине през проверка за удостоверяване на личността преди да тегли пари

ATM Example (with event)



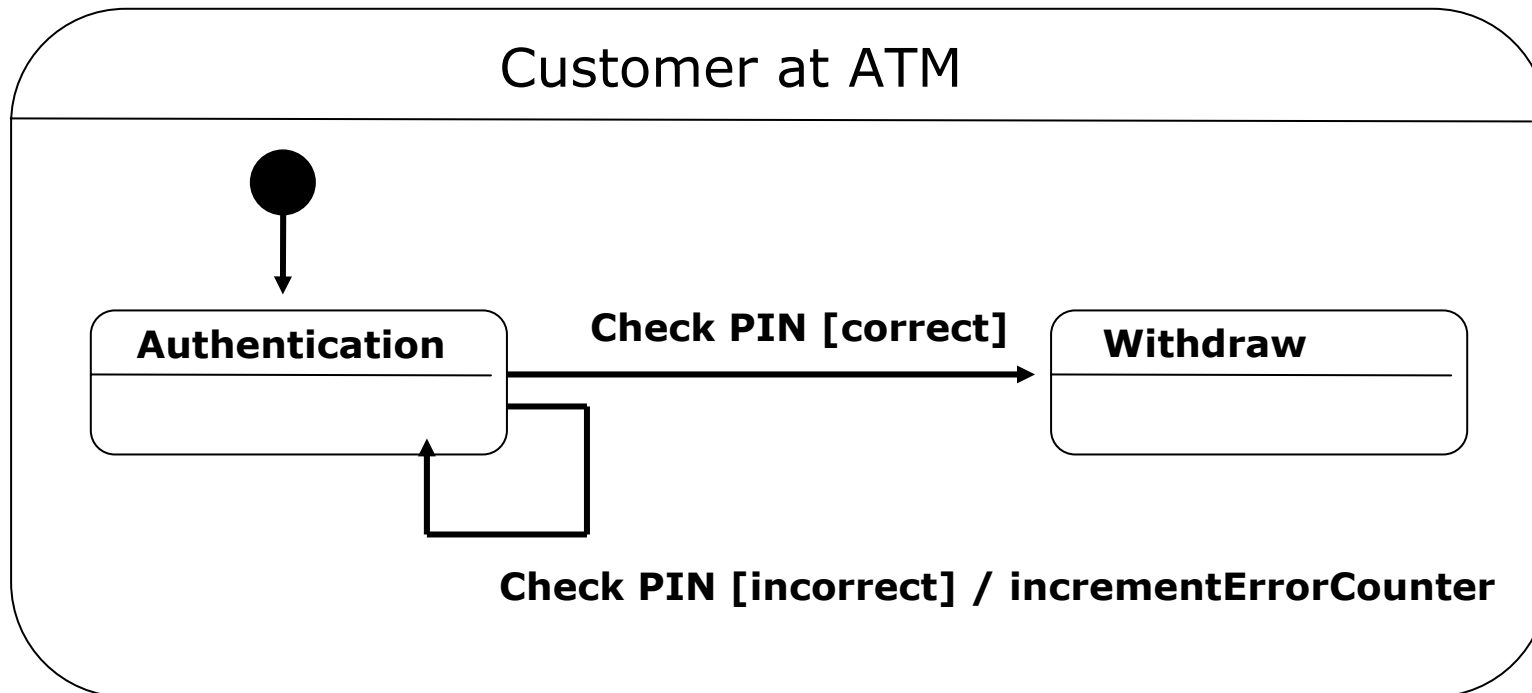
-
- ❑ Удостоверяване на личността - чрез проверка на ПИН.

ATM Example (with guards)



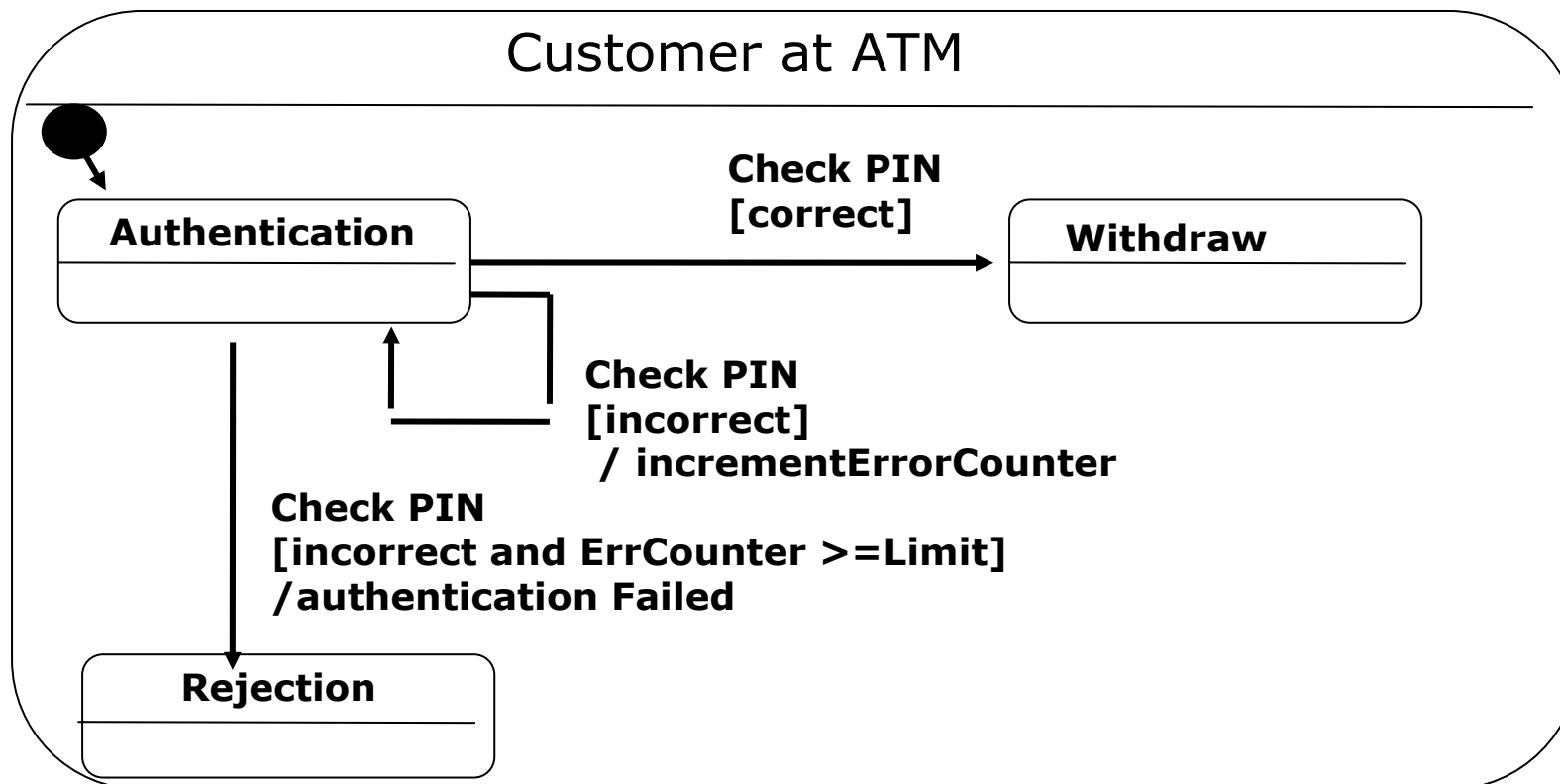
□ Пин код - верен или грешен?

ATM Example (with action)



❑ Неуспешните опити се броят

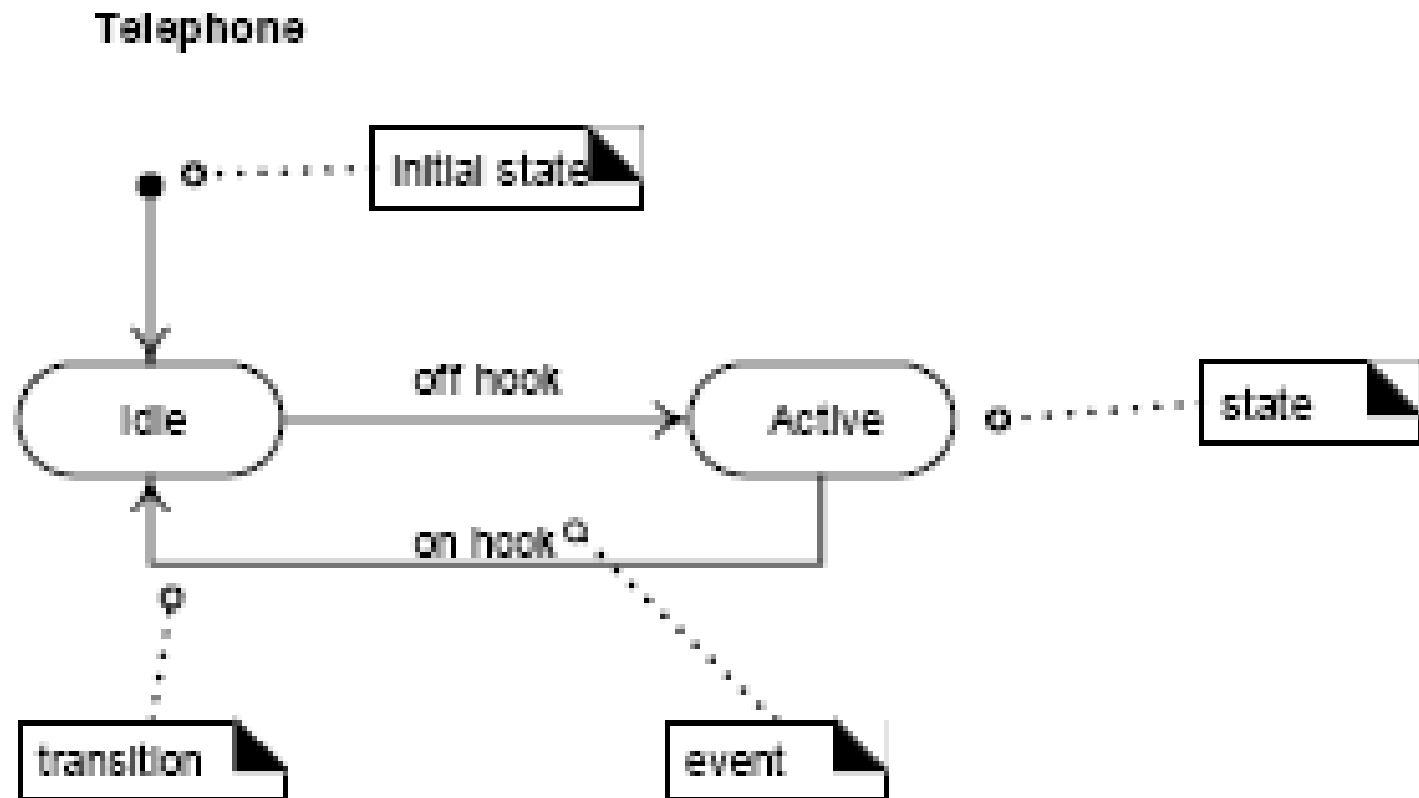
ATM Example (with Send Event)



- Ако броячът надхвърли лимита, потребителят е отхвърлен.

Statechart - пример

Event - A telephone receiver is taken off the hook



Вътрешно събитие (Internal event)

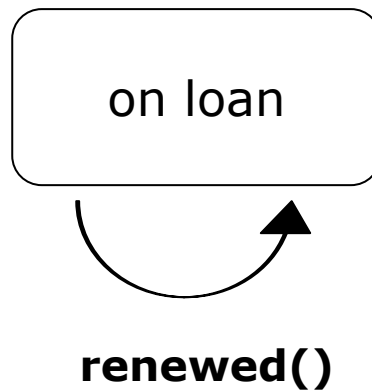
- Вътрешно събитие (**Internal event**)-
причинено от нещо в границите на системата
 - От гледна точка на софтуера, вътрешно
събитие възниква, когато методът е извикан
чрез съобщение или сигнал, който е изпратен
от друг вътрешен обект.
 - Пример:
Когато *Sale* получи *makeLineItem* съобщение, е
настъпило вътрешно събитие
-

Времево събитие (Temporal event)

- **Времево събитие (Temporal event)**— причинено от настъпването на определена дата и час, или откъс от време.
 - От гледна точка на софтуера, времевото събитие е задвижено от реален или симулиран времеви часовник.
 - Пример:
 - След като настъпи операцията *endSale* , в рамките на пет минути трябва да настъпи операцията *makePayment*, в противен случай сегашната продажба е автоматично изчистена.
-

Self-transitions

- Self-transition е преход, чийто изходящи и крайни състояния са едни и същи



Start and stop markers

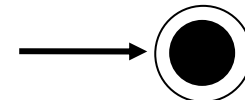
□ Start marker

- Изобразява се като черно кръгче със стрелка в първоначалното състояние на диаграмата.



□ Stop marker

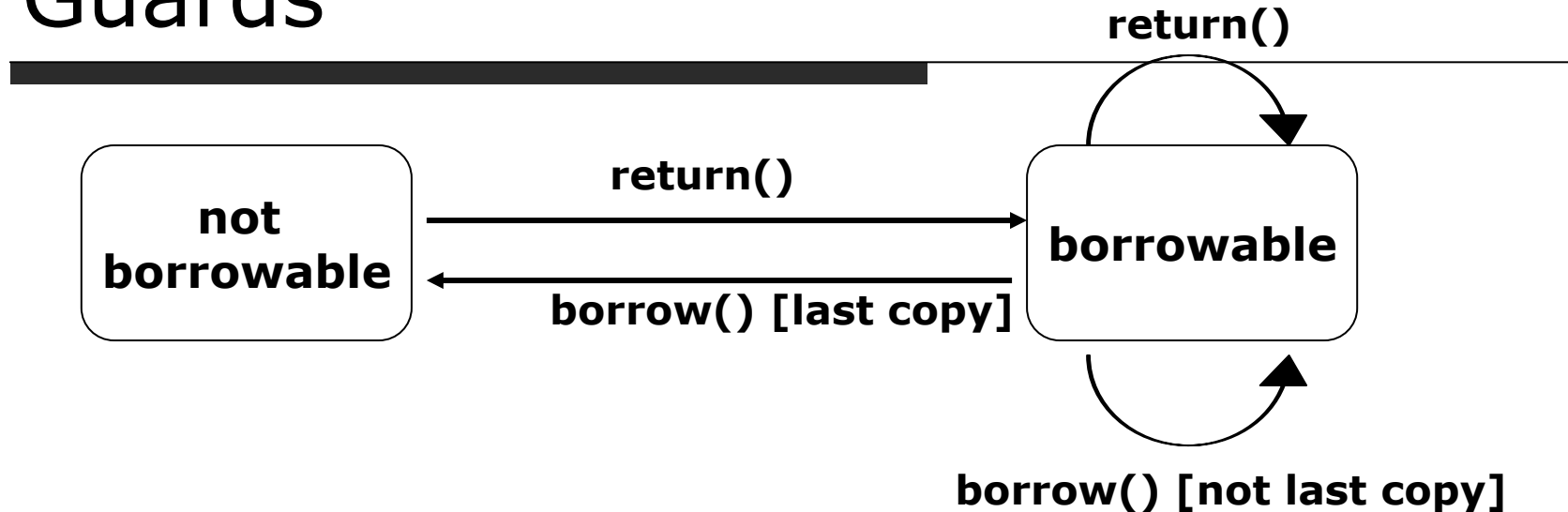
- Изобразява се като черно кръгче с пръстен около него. Крайното състояние означава, че обектът е достигнал края на своя живот.



Guard (guard condition)

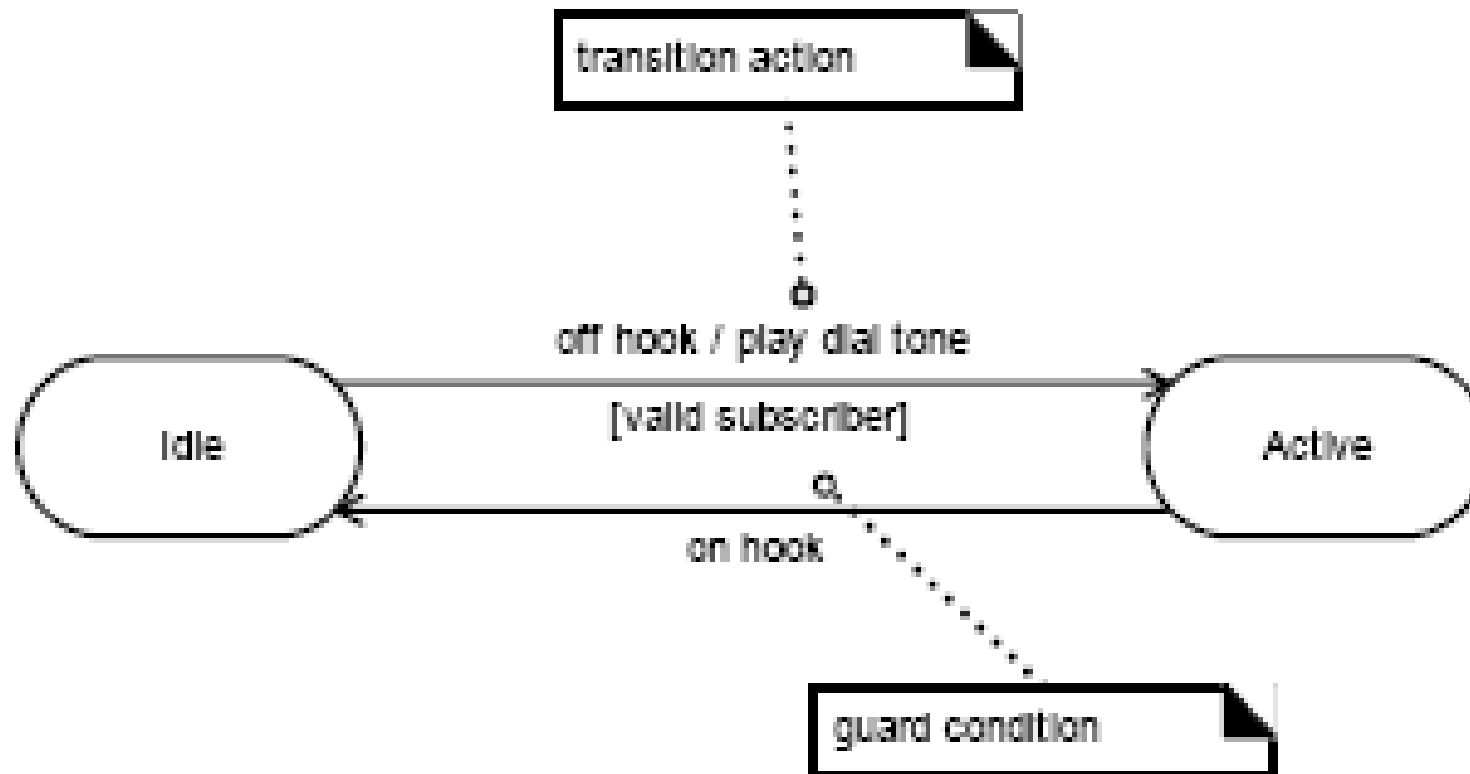
- Guard condition - ограничаващо условие (ако има такова) е булево условие, което трябва да бъде **true**, за да се изпълни преходът
 - Понякога появата на едно и също събитие в едно и също състояние може да предизвика или да не предизвика промяна на състоянието, в зависимост от точната стойност на атрибутите на обекта
-

State diagram for class Book with Guards



- Съобщението *borrowed()* предизвиква промяна на състоянието извън състоянието *borrowable*
 - Само ако това е последното копие на рафта
 - В противен случай книгата-обект остава налична за заемане (*borrowable*)

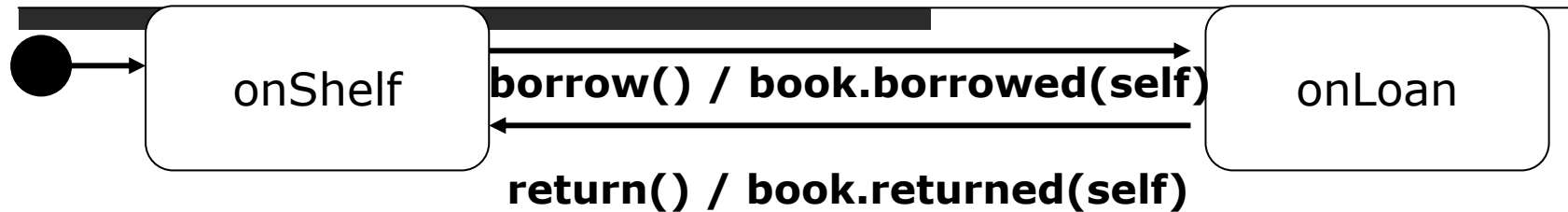
Transition Actions And Guards



Action

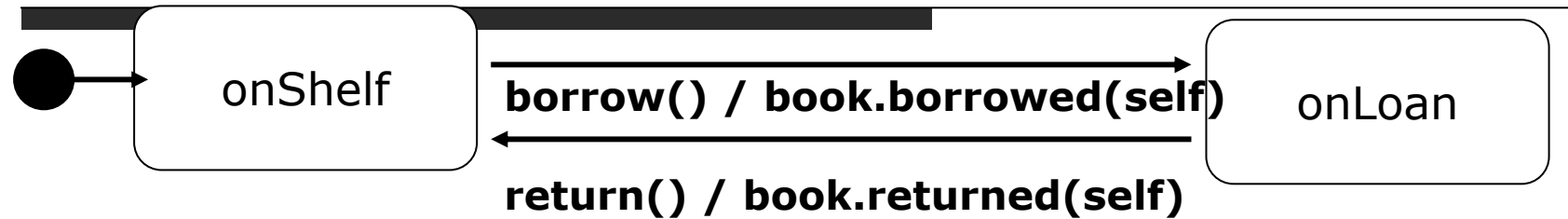
- Действието, което настъпва като резултат от събитие или условие, се изписва след събитието и е предшествано от наклонена черта - "/"
 - Изпращането на съобщение от обект в отговор на получаването на друго съобщение е пример за действие, което представлява реакция на обекта на дадено събитие
 - Например, когато един екземпляр е върнат в библиотека, трябва да бъде изпратено съобщение на класа Book . По този начин класът Book знае колко копия са взети, а вероятно и кои от тях (в зависимост от изискванията)
-

State diagram of class Copy with action



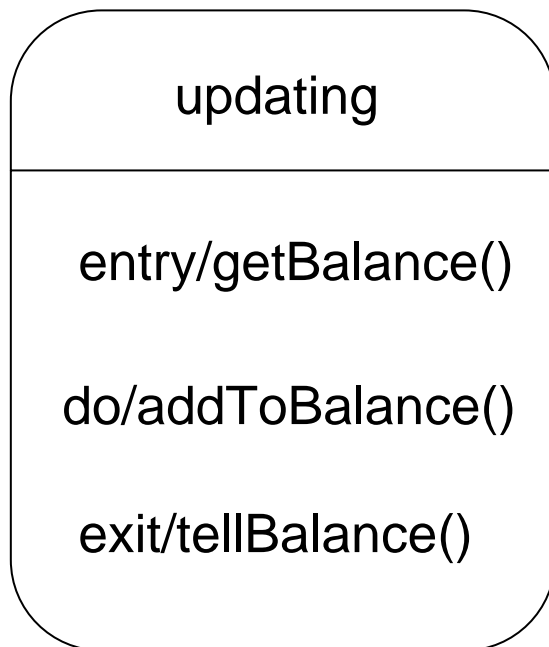
- Събитието `'return()'` задейства съобщение, което променя състоянието на обекта от `'onLoan'` на `'onShelf'`
- Анализирание на нотацията :
 - Наклонена черта (/) показва, че това, което следва, е действие
 - `book`, последвана от една точка идентифицира обекта, към който е изпратено съобщението
 - `returned(self)` е пример за съобщение, включващо параметър, където `self` е обръщане към себе си

How to read this



- Копието е на рафта.
- Когато копието е взето(случва се събитието **borrow**),
 - Действието е да се изпрати съобщение до **book** обекта, че това копие е взето
 - След това се преминава към състоянието **on loan**

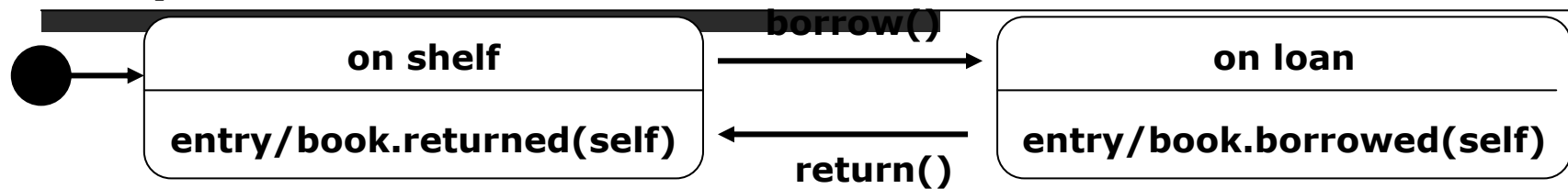
Entry, exit and do actions



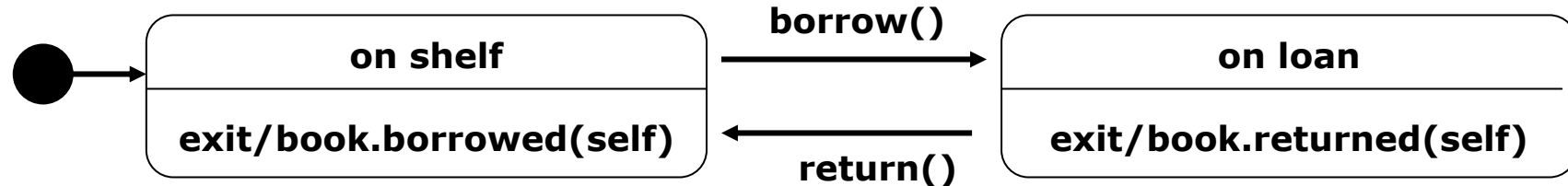
- Както могат да се именуват преходите с действия, така е и възможно да се дефинират действия в рамките на състоянието
 - **entry**
 - определя действие, което трябва да бъде извършено при влизане в състоянието
 - **exit**
 - определя действие, което трябва да бъде извършено, когато се напуска състоянието
 - **do**
 - определя действие, което трябва да бъде извършено, когато е в състоянието
-

State diagram of class Copy with entry/exit actions

Entry action



Exit action



- Ние можем да покажем нашето намерение пряко, пишейки действието вътре в състоянието, като реакция на специалното събитие (например *entry* или *exit*)

