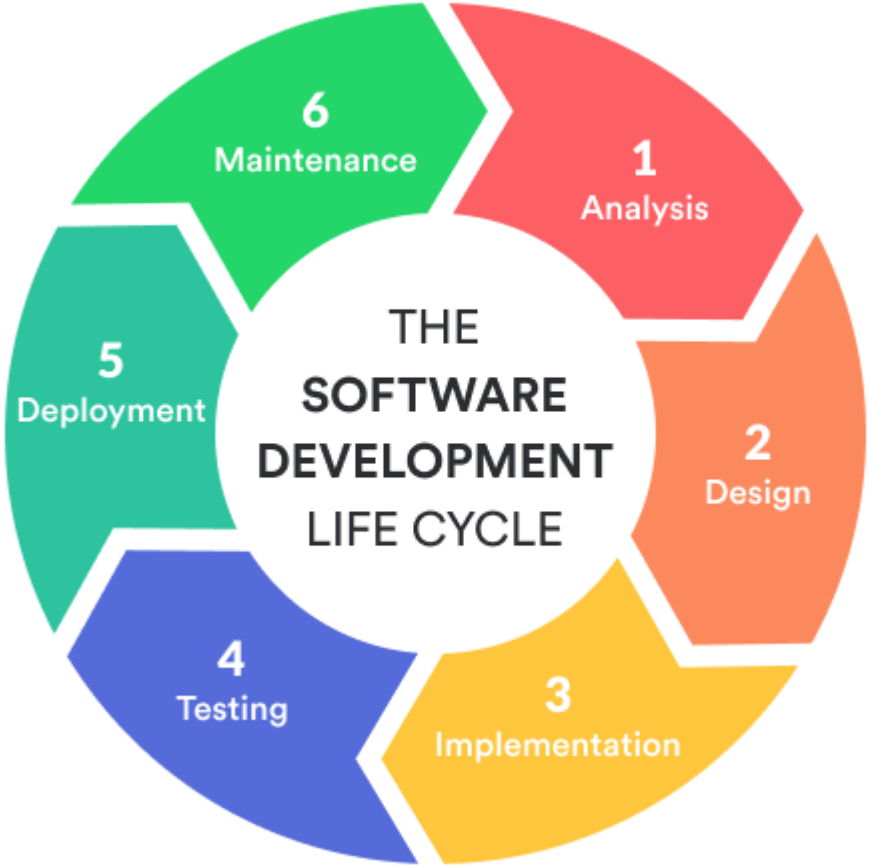
Inxhinieria Sistemeve Softuerike

Modelimi/Dizajni i Sistemit:

FAKULTETI: SHKENCA KOMPJUTERIKE DHE INXHINIERI

Kalimi nga kërkesat në Dizajn apo Modelim

Ri-kujtim: Cikli i Jetës së Zhvillimit të Softuerit



UML – Diagrami i Aktivitetit

Për **UML-in**: Analiza dhe dizenjimi i **orientuar në objekte**

- **Gjuha e modelimit e unifikuar (UML)** është një ***standard i simboleve në industri*** për ***dizajnimin e sistemeve*** të ***orientuara në objekti***, i mbështetur nga Grupi i Menaxhimit të Objekteve (OMG).

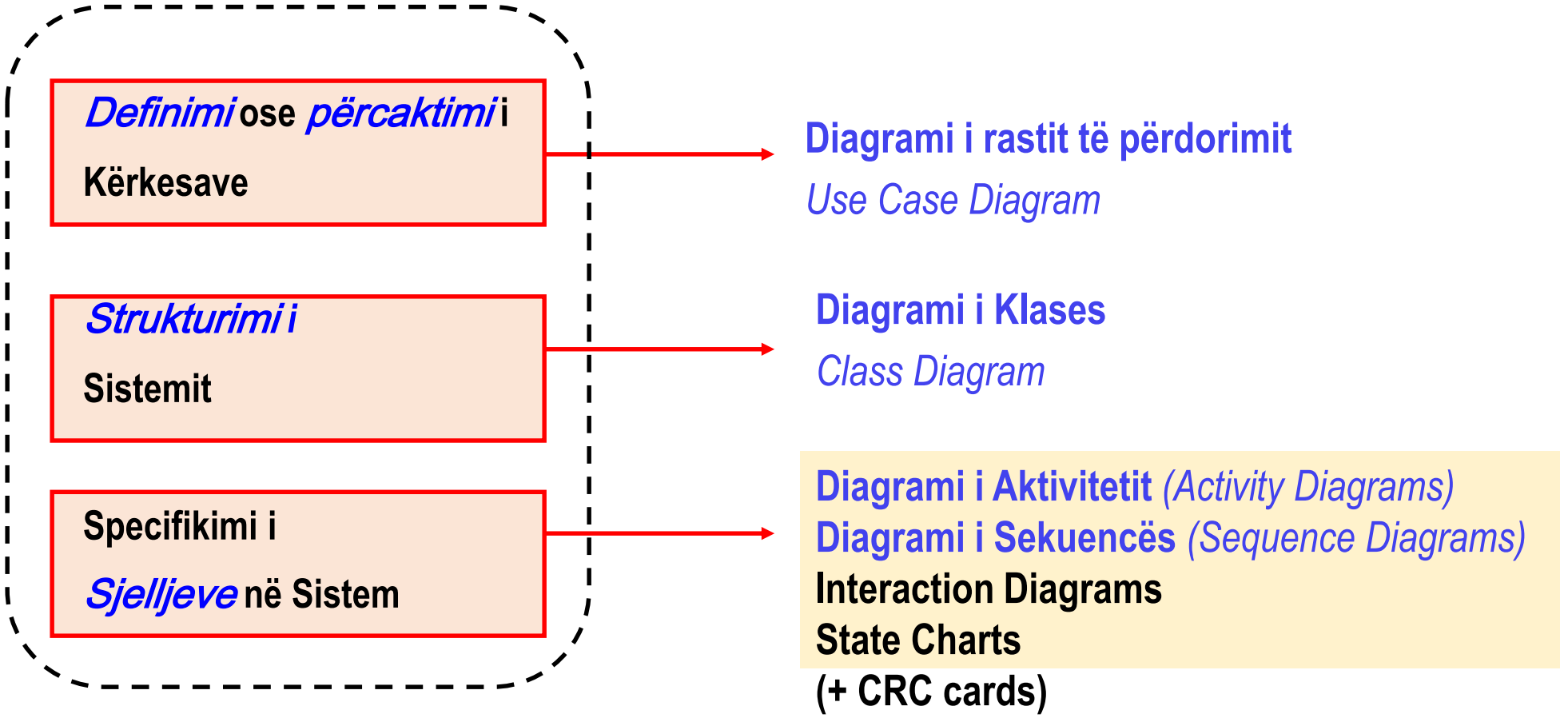
- **UML-i** është duke u adoptuar gjerësisht në *shumë fusha të* ***softuerit*** dhe **komunitetit** të zhvillimit të **sistemeve kompjuterike**.

Një **vështrim i përgjithshëm** i **UML** dhe **komponentëve** të tij:

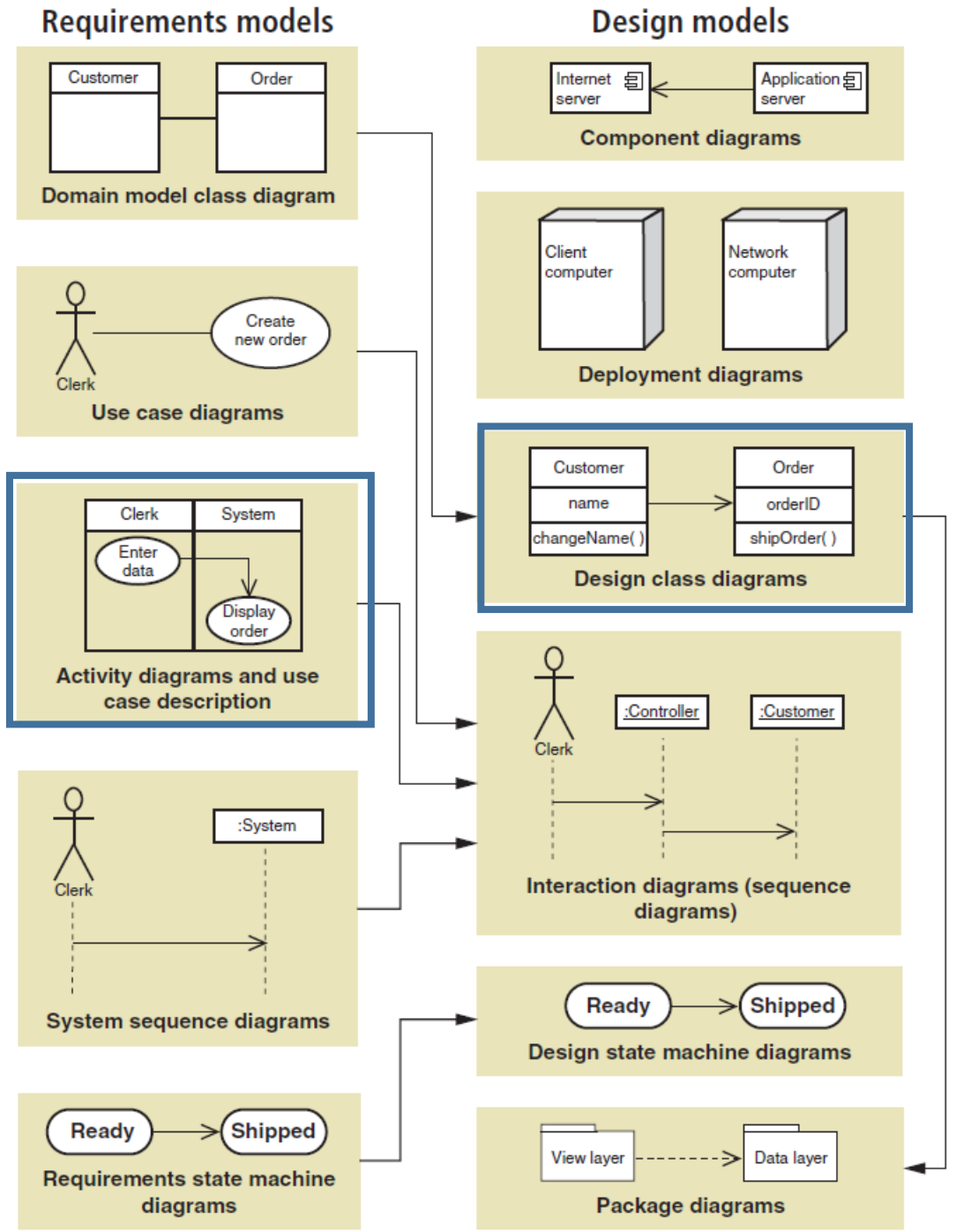
- **Gjërat (Things),**
- **Relacionet (Relationships),** and
- **Diagramet (Diagrams)**

UML Category	UML Elements	Specific UML Details
Things Gjërat	Structural Things	Classes Interfaces Collaborations Use Cases Active Classes Components Nodes
	Behavioral Things	Interactions State Machines
	Grouping Things	Packages
	Annotational Things	Notes
Relationships Relacionet	Structural Relationships	Dependencies Aggregations Associations Generalizations
	Behavioral Relationships	Communicates Includes Extends Generalizes
Diagrams Diagramet	Structural Diagrams	Class Diagrams Component Diagrams Deployment Diagrams
	Behavioral Diagrams	Use Case Diagrams Sequence Diagrams Communication Diagrams Statechart Diagrams Activity Diagrams

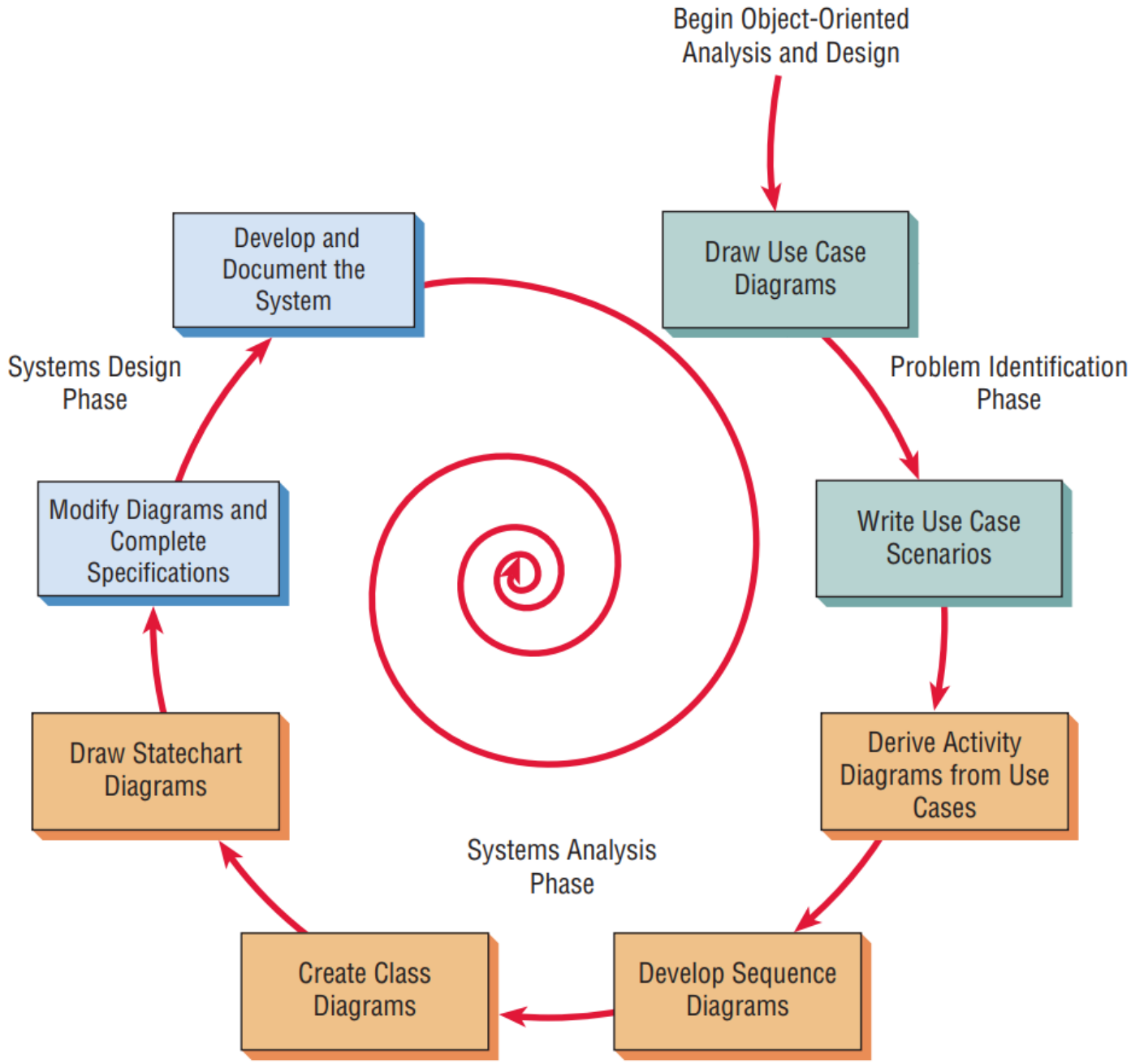
Diagramet kryesor në UML



UML Requirements vs. Design Models

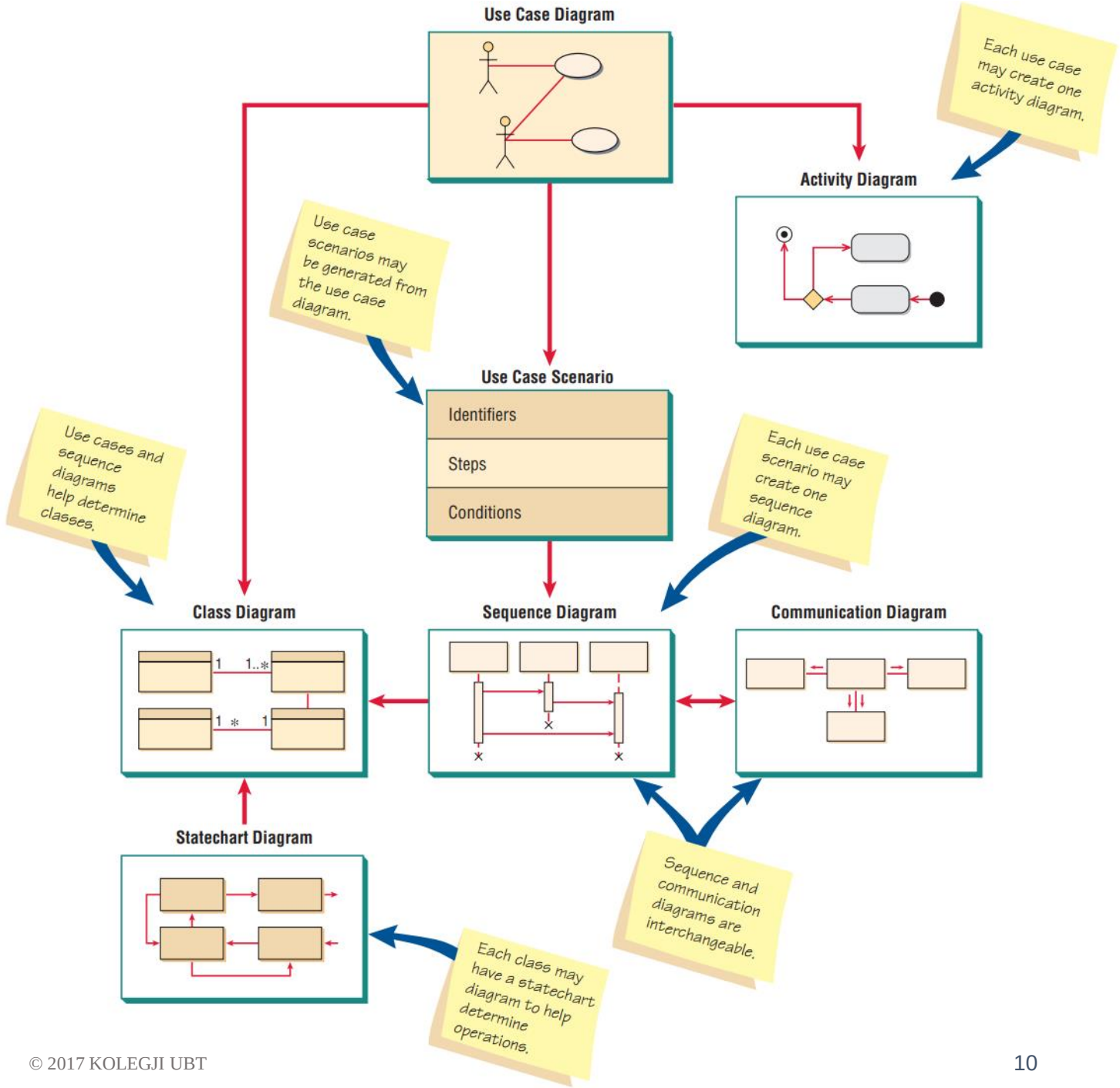


Hapat në procesin e zhvillimit në UML.



Një pamje e përgjithshme e diagrameve të UML-it

Paraqet se si diagramet e ndryshme në UML **ndërlidhen** në **implementimi** e diagrameve në UML



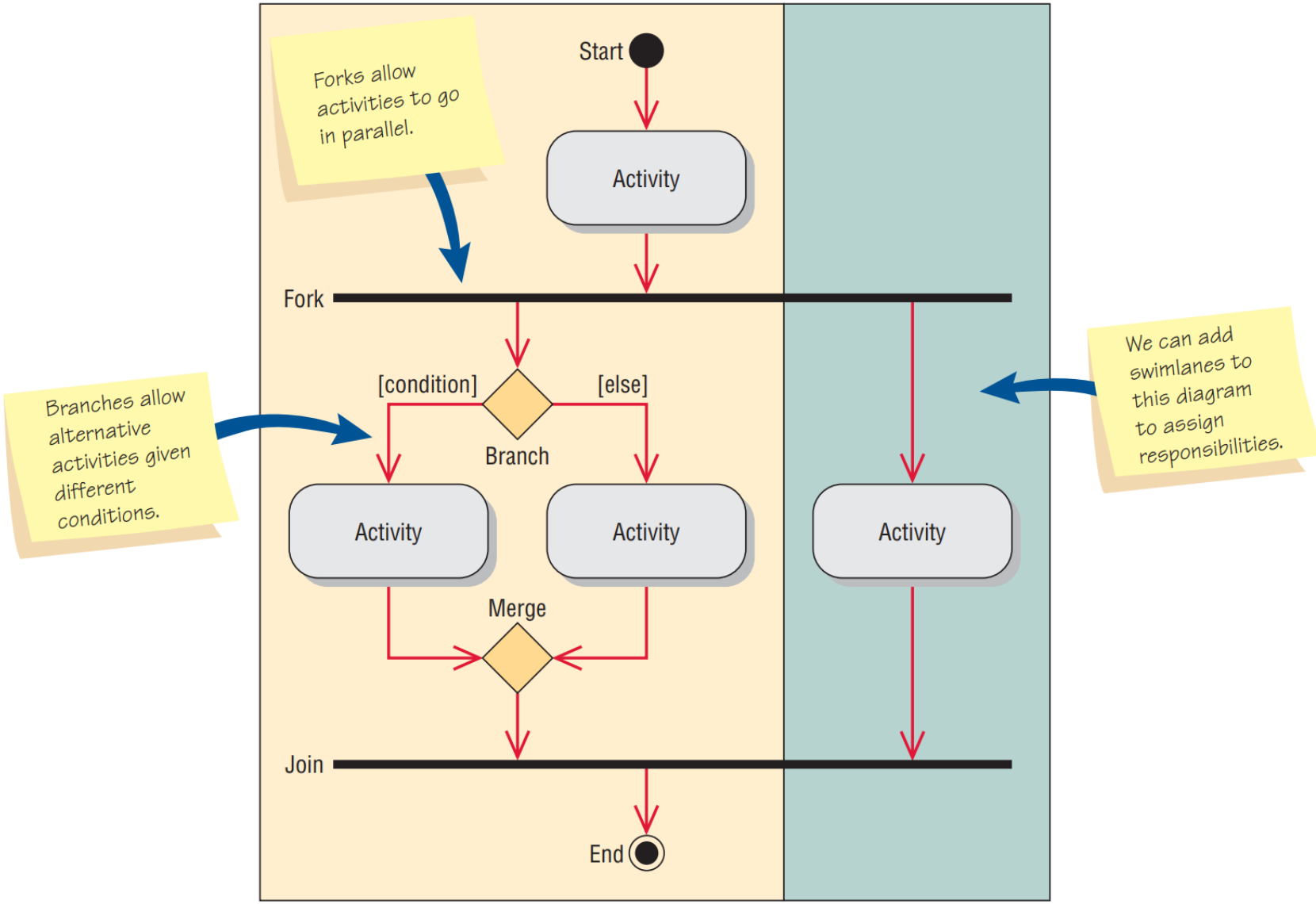
Diagramet e Aktivitetit

- **Arsyeja** kryesore për të përdorur **diagramet e aktivitetit** është të *modelojë rrjedhën e punës* të sistemi që është duke u **dizajnuar**.
  - **Diagramet e aktivitetit** tregojnë **renditjen e aktiviteteve** në një *proces*, duke përfshirë aktivitetet *sekuenciale* dhe *paralele*, dhe *vendimet (kushtet)* që bëhen.
  - Një **diagram aktiviteti** zakonisht krijohet për *një rast të përdorimit* dhe mund të tregojë *skenarë të ndryshëm* të mundshëm.

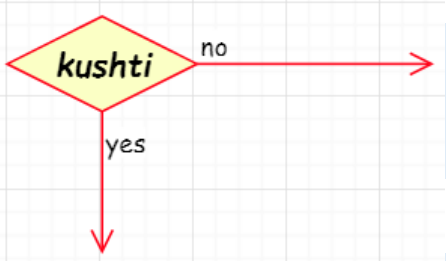
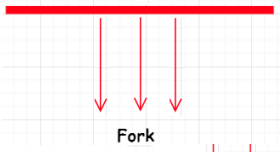
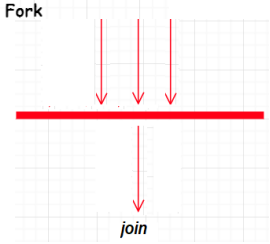
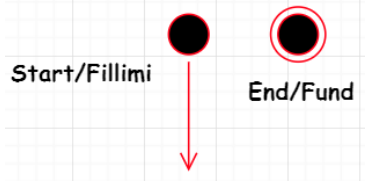
→**Diagramet e aktivitetit** *nuk duhet të zëvendësojnë* diagramet e *ndërveprimit* dhe diagramet e *gjendjës*.

→**Diagramet e aktivitetit** *nuk japin detaje* se **si sillen objektet** ose se si *bashkëpunojnë* **objektet**.

Simbolet në një diagram të aktivitetit



Simbolet në një diagram të aktivitetit

Simbolet	përshkrimi
	Një **aktivitet** paraqet, ose **një proces manual**, siç është **nënshkrimi i një dokumenti ligjor** ose një **proces automatizuar**, si një **metodë** ose një **program**.
	Një **shigjetë** përfaqëson një ngjarje. Ngjarjet përfaqësojnë gjërat që ndodhin në një kohë dhe vend të caktuar. Shigjeta përfaqësojnë **drejtimin e rrjedhës** në diagram. Pikat drejtimi i shigjetës paraqet/ tregon progresin e aktiviteteve
	Një **diamant** përfaqëson ose **një vendim** (i quajtur edhe **një degë**) ose një **bashkim**. Vendimet kanë **një shigjetë** që **hyn** në diamant dhe **disa që dalin**. Mund të përfshihet një kusht që tregon vlerat e **gjendjes**. Bashkimet tregojnë disa ngjarje që kombinohen për të formuar një ngjarje.
	Një **drejtkëndësh i rafshët** dhe i gjërë përfaqëson një **shirit të sinkronizimit**. Këto përdoren për të treguar **aktivitete paralele**, dhe mund të kenë një ngjarje që **futet** në **shiritin e sinkronizimit** dhe **disa ngjarje që dalin prej saj**, të quajtur një **pirun (FORK)**
	Një **sinkronizim** në të cilin **disa ngjarje** që **shkrihen/rezultojn** në **një ngjarje** quhet një **bashkim (JOIN)**
	Ka dy simbole që tregojnë **fillimin** dhe **fundin** e diagramës. Fillimi shfaqet si një **rreth i mbushur**. Gjendja **përfundimtare** shfaqet si një **rreth i zi i rrethuar** nga një rreth i **bardhë**.

Gjeni sjelljet: e një rasti

Me një partner vizatoni një diagram për të kapur procesin e mëpos
(sikur ta përshkruani në mënyrë grafike):

“Ju jeni duke fjetur në shtrat, priteni për alarmin tuaj. Kur përfundon alarmi ju zgjoheni, vishuni dhe zbriseni në katin e poshtëm. Ju pregaditeni pak mëngjes dhe përderisa ju hani mengjesi ju gjithashtu lexoni gazetën e mëngjesit. Kur të mbaroni, largoheni nga shtëpia.”

Shikoni në diagramin tuaj (1)

- Cilat ishin karakteristikat e diagramin tuaj?
  - Veprimet ose Ngjarjet (Actions or Events)
  - …

Shikoni në diagramin tuaj (2)

- What were the features of your diagram?

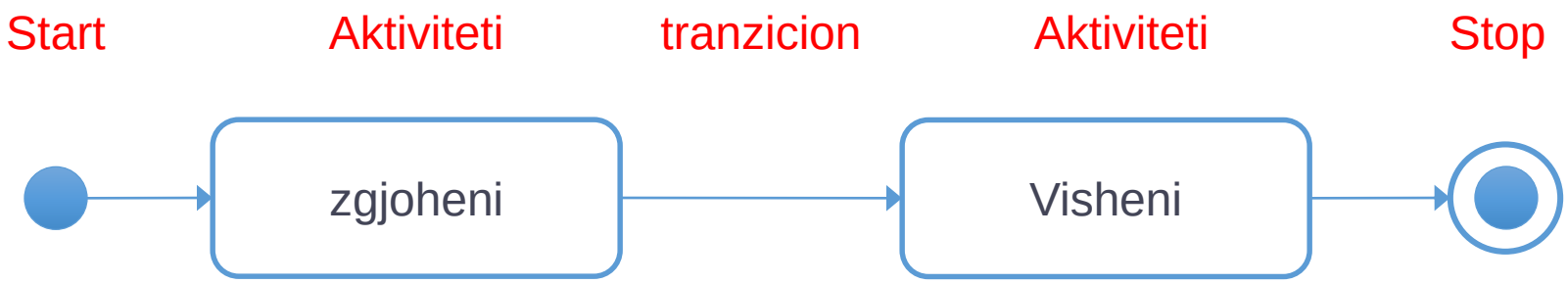
  - Veprimet ose Ngjarjet (Actions or Events)
  - Sekuenca
  - Paralele
  - Një Start dhe Stop
  - Vendimet'kushtet
  - Ndarjet logjike?

UML-Diagramet e Aktivitetit (1)

- UML ekuivalent me një FlowChart
  - Ofron një pamje të nivelit të lartë të asaj që po ndodh *brenda* një Rasti të Përdorimit (use Case)
  - (Është një variant i një Diagrami të gjendjës në UML)

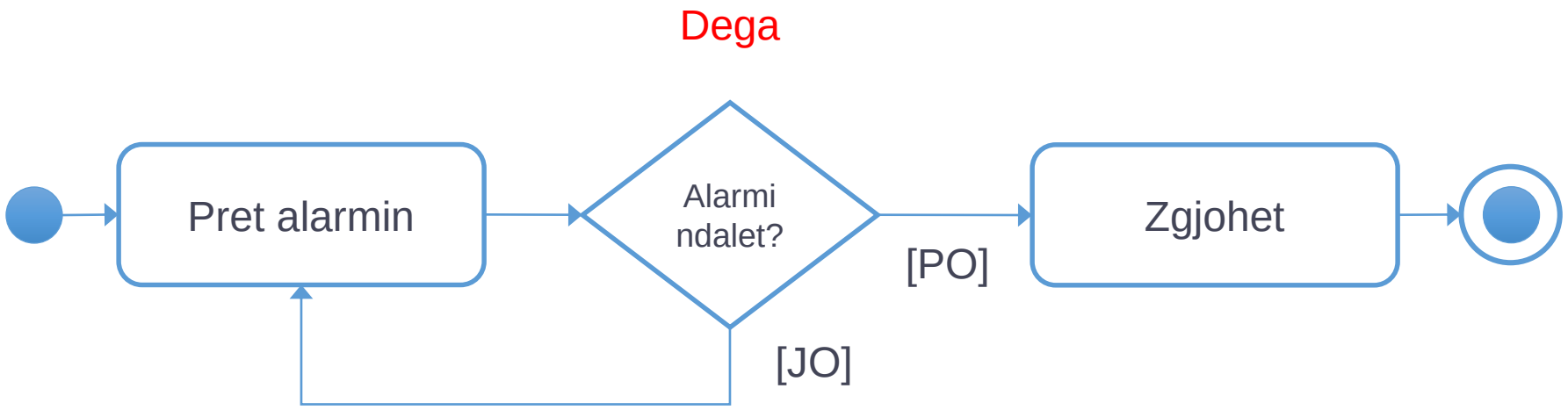
UML-Diagramet e Aktivitetit (2)

- Është e bazuar në:
  - **Aktivitetet**
  - Lidhur me **tranzicione**
  - Me një **Start** dhe një **Stop**



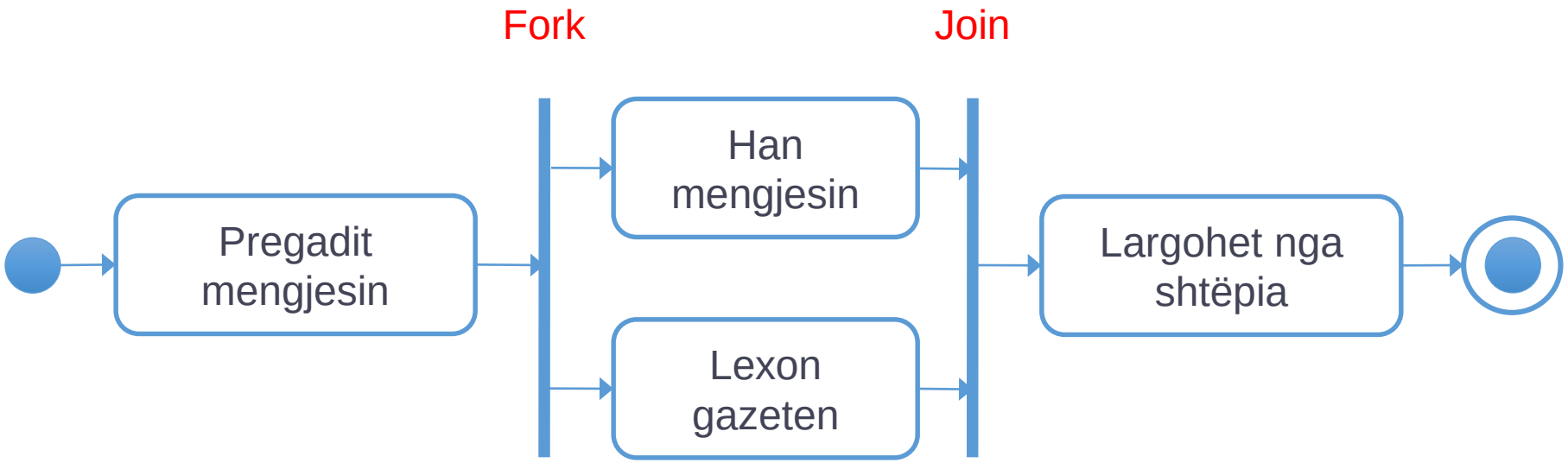
UML-Diagramet e Aktivitetit (3)

- Pikat e vendimeve (degët) janë shënuar me simbolin e diamantit
  - Një **degë** ka një përshkrim ***opsional***
  - Tranzicionet nga ***një dege*** janë **emërtuar** (kushtet)

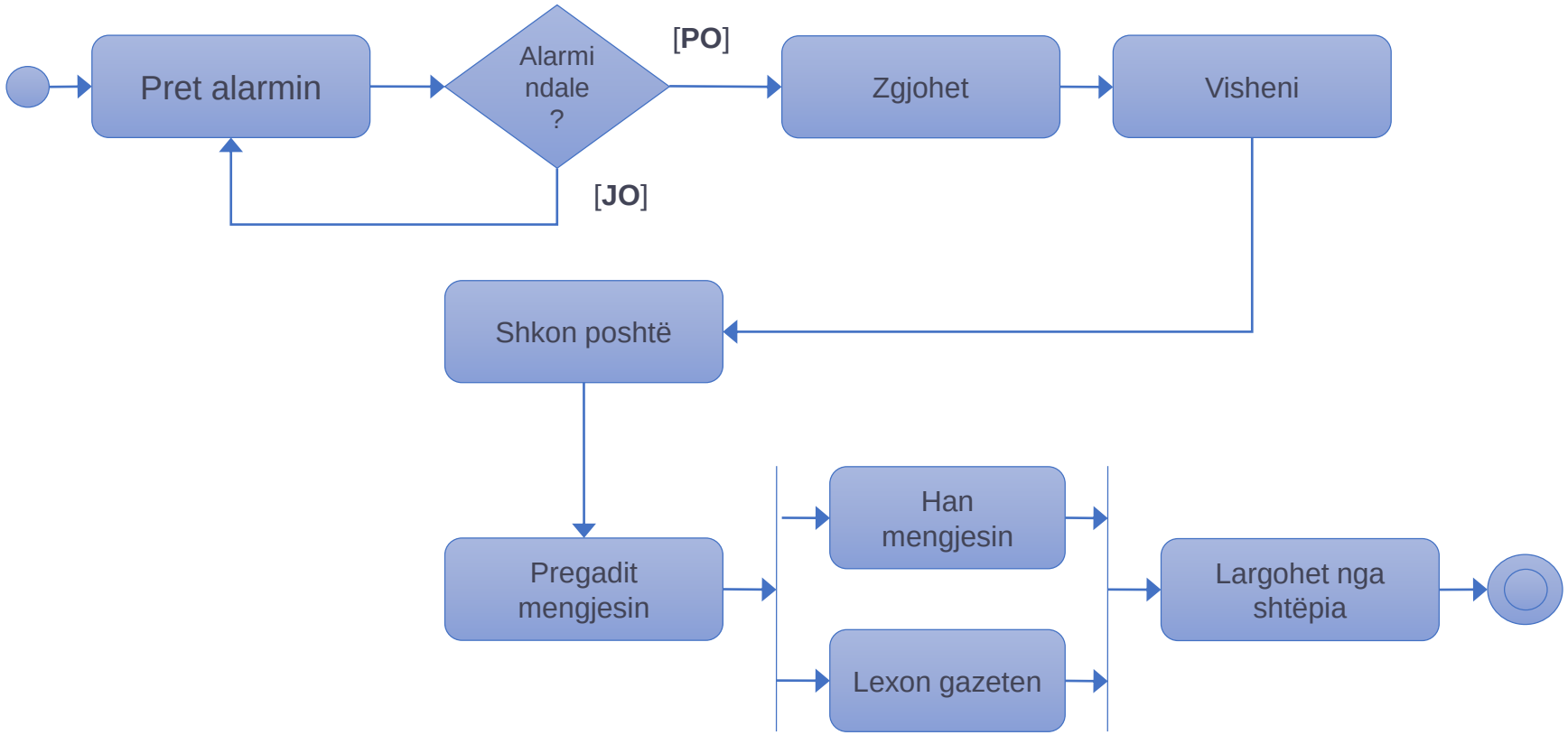


UML-Diagramet e Aktivitetit (4)

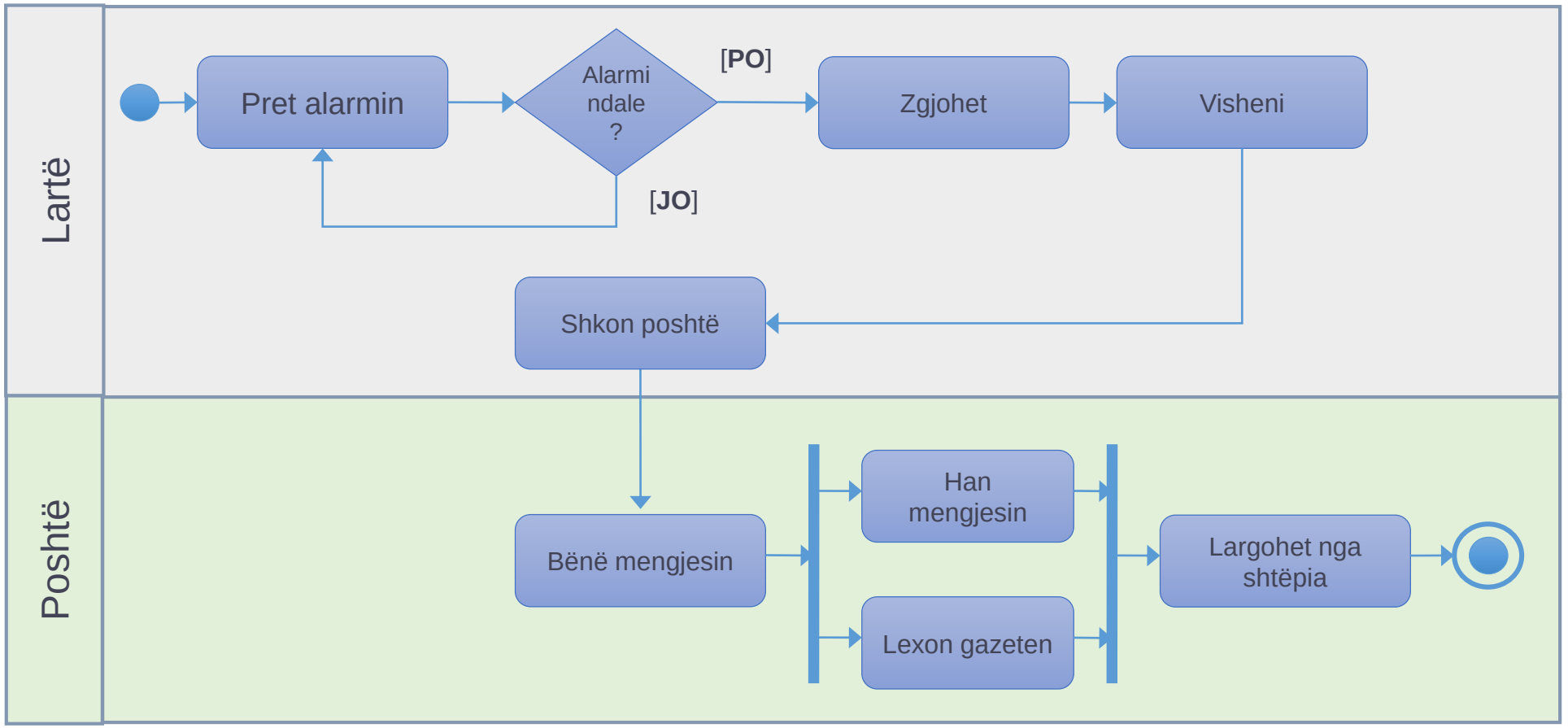
- Shiritat (Forks dhe Joins)
  - Përdoret për të treguar si gjërat ndodhin paralelisht
  - Ose për të sjellë së bashku disa tranzicione



Vendosim të gjithë së bashku



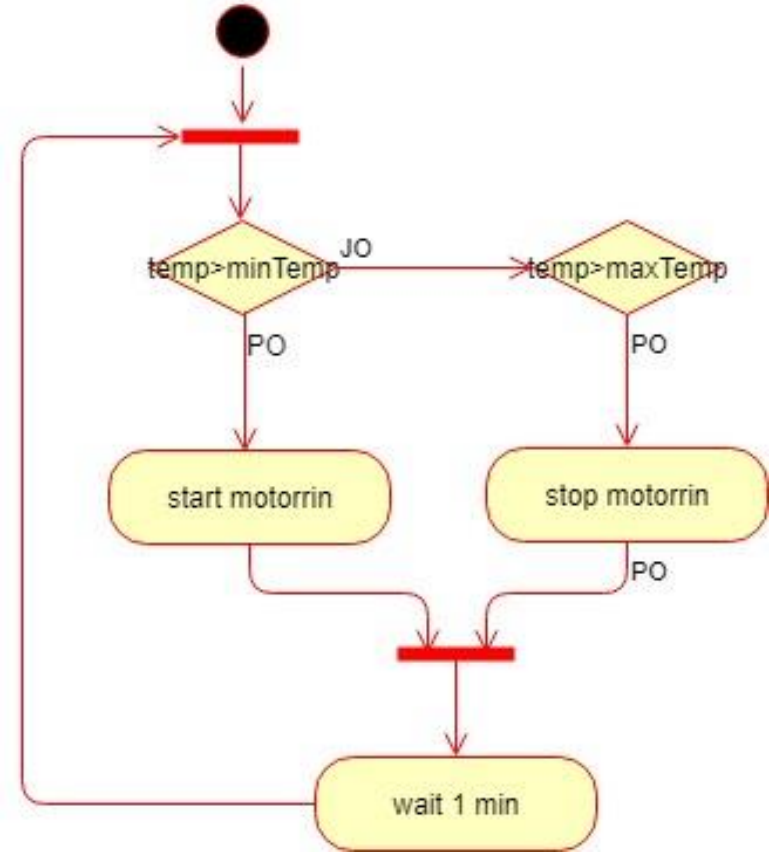
Shtoni: Swimlanes (1)



Shtoni: Swimlanes (2)

- Swimlanes ndajn një diagram
  - Përdoret për të treguar **fusha** të ndryshme **logjike**
- Diagramet shpesh mund të ndahen në mënyra të ndryshme
  - Sipas një **Faze**
  - Sipas **Aktorit**
  - Sipas **Departamentit**
- Nuk ka asnjë mënyrë e drejtë
  - Ndarja në çdo gjë që është mënyra më e dobishme

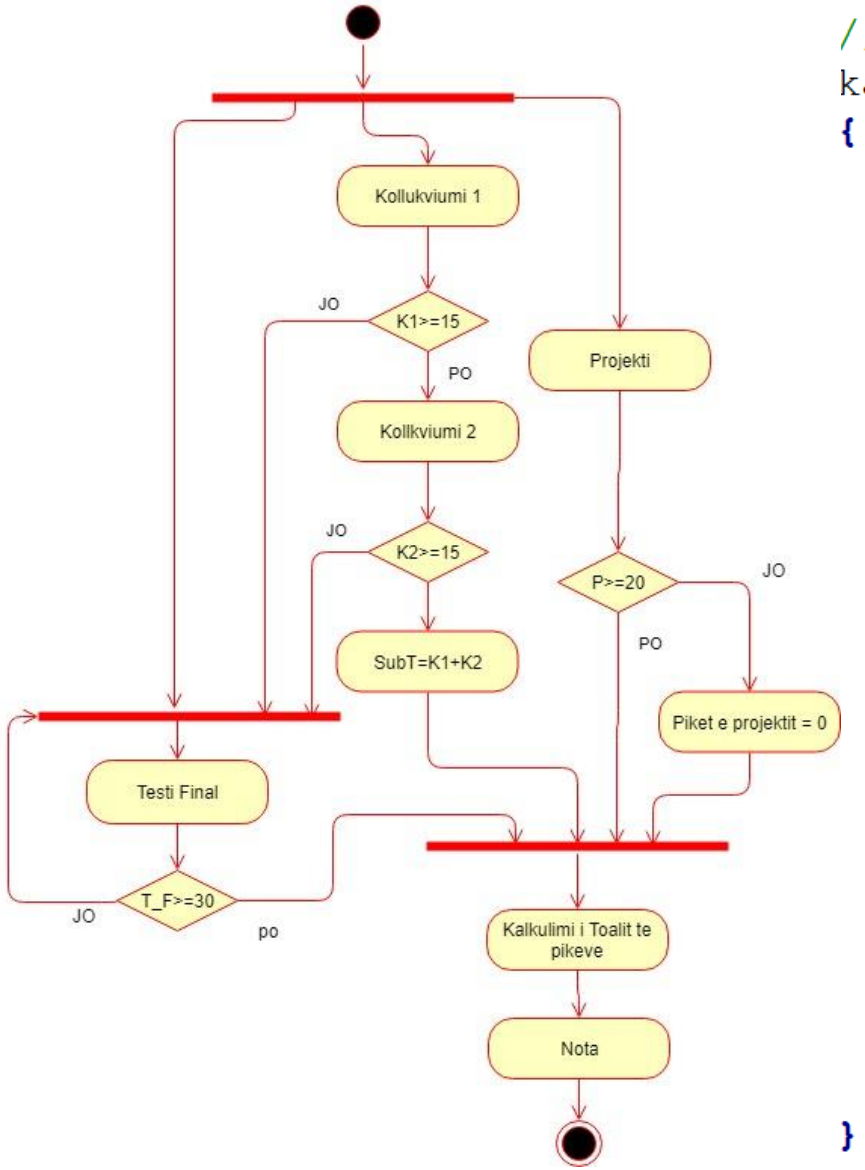
Shembull 1: Konvertimi i *diagramit të aktivitetit në Psudokod*




```
while (running)
{
      kontrollo temp
      if (temp > minTemperatura)
            start motorrin
      else if (temp >= maxTemperatura)
            stop motorrin
      wait 1 min
}
```

Shembull 2: *Kalkulimi i notes* në Lëndën Inxhinieria Softuerike

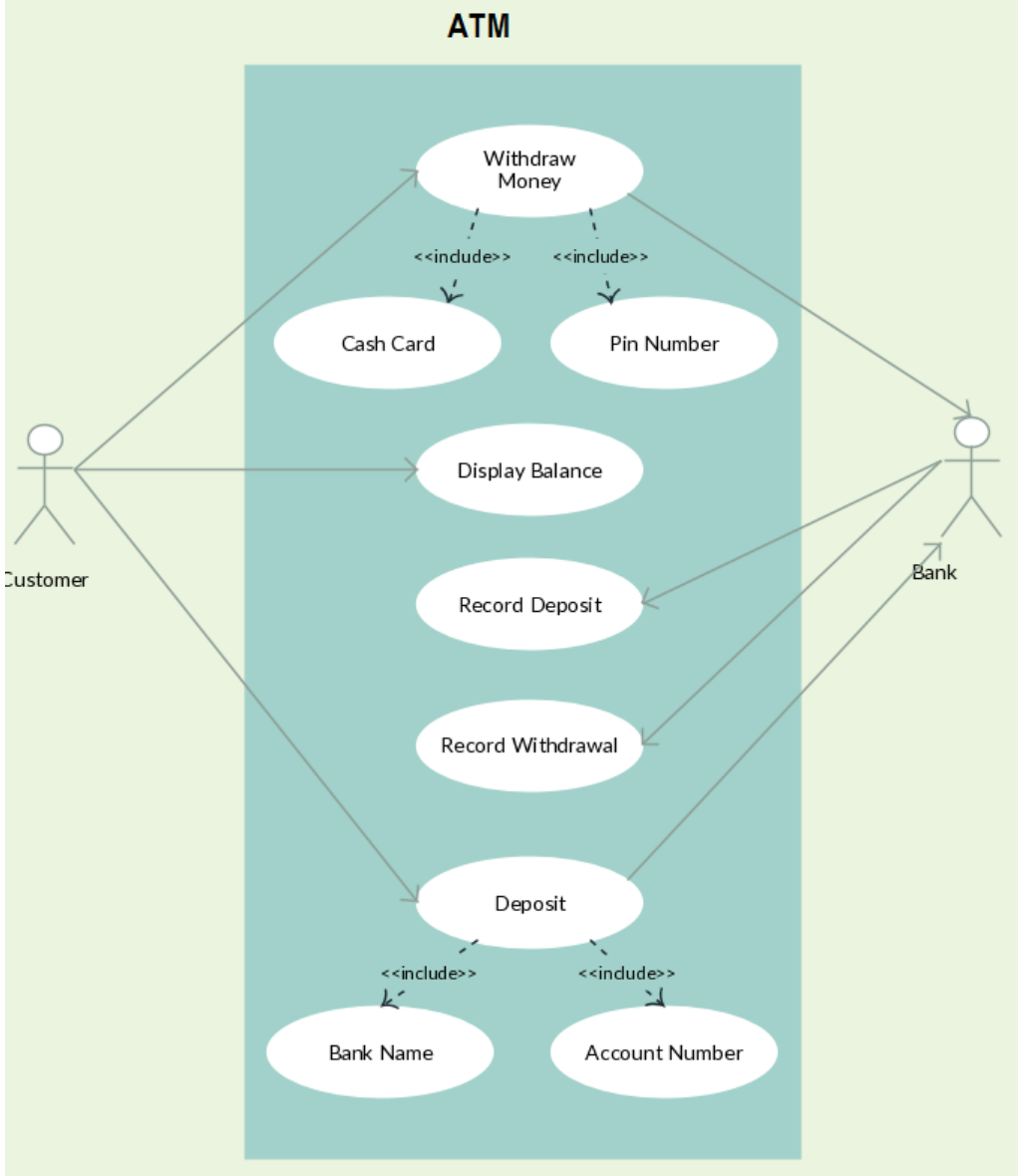
UC_ID : UC_10
Titulli/UC EMIR : ***Kalkulimi i notes në Lëndën Inxhinieria Softuerike***
AKTORI PRIMAR: **Profesori (RAMIZ HOXHA)**
PËRSHKRIM I SHKURTËR: **Profesori** kalkulon ***noten*** për lëndën **Ixh.Sof** duke i shtuar të dhënat e aktivitetev përkatëse
KUSHT PARAPRAK: Studenit duhet të ketë prezencen në ligjerata dhe ushtrime valide (>=75%)
TRIGGER/SHKAKU: Profesori follon kalkulimi e notës
SKENARIA KRYESORE (I SUKSESIT): Profesori shton pikët e kollekviumit_1, nëse pikët e kollokviumit_1 të studentit ka arritur kushtin e kualfikimit (kaluse, d.m.th K1>=15 pikë). Ather profesori shton pikët e kollekviumit_2 po ashtu edhe në këte K2 student duhët të i arri min 15 pikë (K2>=15 pikë). Profesori shton pikët e projektit ku nëse janë arritur min 20 pikë (P>=20 pikë) atëher student kulifikohet për kalkulim të totalit të pikëve duke vlersuar në notën perkatës. Po ashtu student nëse nuk ka pasur sukses në K1 dhe K2 studenti hynë direct në test final. Po ashtu në tesitn final student duhët të i arrij min 30 pikë (TF>=30 pikë) pastaj këto pikë i shtohen edhe pikët e projekti të studntit pastaj kalkulohet nota. Totali duhet të jetë min 50 pikë. Përndryshen hyn në afatin e ardhshëm të provimit.




```
//Psudocodi kalkulimi i notes SWE
kalkulo_Noten_SWE()
{
     lexo (K1 ose TF ose P)
     if (K1>=15)
         Temp1=K1
     lexo K2
     else
         lexo TF
     if (K2>=15)
         subT=Temp1+K2
     else
         (lexo TF ose lexo P)

     if (TF >=30)
        TF+=TF
        else set_to TF=0
      Lexo P
     if (P>=20)
         P +=P
     else P=0
     Totali=TF+P ose Totali=SuT+P
     return nota = Nota: Totali
}
```

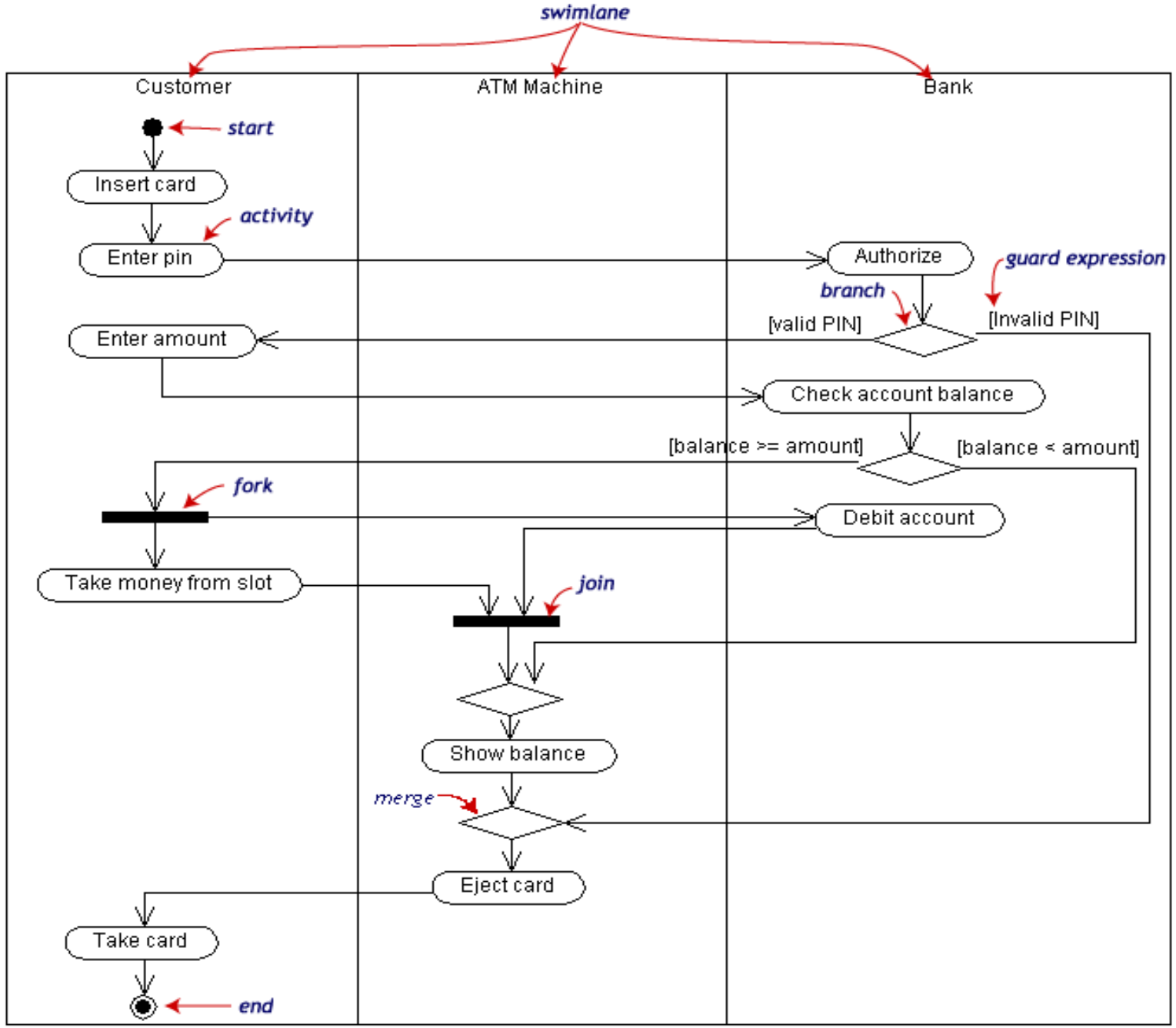
Shembull 2: *Tërhiqni para nga një llogari bankare nëpërmjet ATMs (rasti i përdorimit)*



UC_ID : UC_02
TITULLI/ UC EMRI***: Tërhiqni para nga ATM (klienti ynë bankar)***
ACTORI KTYESOR: ***Klienti ATM/Bankomat (Kosumatori)*** AKTORIT E DYTË: ***ATM (BANKOMATI), Banka***
PËRSHKRIM I SHKURTËR: ***Klienti ATMs tërheq fonde nga një makinë ATMs***
PARAKUSHTI: ***Klienti ynë bankar duhet të ketë kartë bankare***
TRIGGER: ***Përdoruesi fillon transaksionin duke futur një kartë bankare***
SKENARIA KRYESORE: 1. Klienti fut kartën në ATM. 2. Klienti shton PIN. 3. Banka validon PIN (Autorizon). 4. Klienti shton/fut shumën e dëshiruar 5. Bank verivikon Bilancin e llogarisë 6. Klienti merr para nga hapsir/slot 7. Llogaria e debitimit përditësohet 8. ATM shfoq Bilancin e llogarisë 9. ATM nxjerr jasht Karta e kreditit 10. Klienti merr kartën 11. Rasti i përdorimit përfundon.
ZGJERIMI (EXTENSIONS): (Shënim: shpesh zgjerimet janë komplekse.) 3a. Nëse PIN i pavlefshëm “Nxjerr kartën" (shiqo UC_RejectCard) 5a. Nëse Klienti nuk ka fonde të mjaftueshme, ATM tregon Bilanci që klienti do të mund të tërheqë dhe Skenari rifillohet në hapin 4. 5b. Nëse Klinto ka më pak para në llogarinë e tyre sesa shuma minimale që mund të financojë makina, atëherë ATM shfaq një mesazh që tregon se karta është nxjerrë.

Activity diagram for:

***withdraw money** from a **bank account** through an **ATM** (use case)*



Mangësitë

- Një disavantazh i diagrameve të aktivitetit është se ata nuk paraqesin në mënyrë *eksplicite cilat objekte ekzekutojnë* ***cilat aktivitete***, dhe ***mënyra se mesazhi funksionon mes tyre***.
  - Etiketimi i çdo aktiviteti me objektin përgjegjës mund të bëhet.
  - Është e dobishme të vizatoni një diagram aktiviteti në fillim të modelimit të një procesi, për të kuptuar procesin e përgjithshëm.

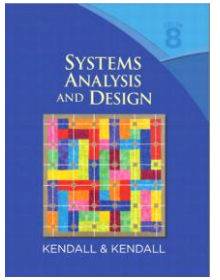
- Pastaj diagramet e ndërveprimit mund të përdoren për t'ju ndihmuar të shpërndani aktivitetet në klasa.

Përmbjedhje

- UML **Diagramet e aktivitetit** modelim i ***kontrollit të rrjedhjës***
  - **D**iagramet e **A**ktivitetit përfshijnë:
    - **Degëzimi** (***vendimmarrja***)
    - **Piruni** (fork) dhe **Bashkim** (Join) (aktivitete ***paralele***)
    - **Swimlanes** (***ndarje*** logjike)
    - Kur **analizohet** një ***rast përdorimi***
    - Cilat ***veprime*** ekzistojnë dhe ***kur ndodhin***?
    - Ky diagram nganjëherë quhet i ***kontrolli i rrjedhjës***

Referencat

- **Kapitulli 10**: Systems analysis and design 8 Ed. By Kenneth E. Kendall



- **Kapitulli 5**: Software Engineering. 9th ed. By Ian Sommerville