



Концептуално пројектовање: Дијаграм класа података

Пројектовање база података

Василије Тодоровић, Анђела Дамњановић, Александар Стефановић

7. април 2026.



Садржај

- 1 UML дијаграм класа
- 2 Пример - издавачко предузеће
- 3 Пример - организовање музичких концерата
- 4 Додатак



Садржај

- 1 **UML дијаграм класа**
- 2 Пример - издавачко предузеће
- 3 Пример - организовање музичких концерата
- 4 Додатак



UML дијаграм класа

- Поред већ споменутог ER дијаграма, за конструкцију концептуалног модела може се користити и дијаграм класа података.
- UML дијаграм класа већ описује структуру класа података. Идеја је да се исти дијаграм искористи за моделирање података.
 - У првом плану су атрибути и односи, понашања се занемарују.



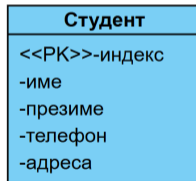
UML дијаграм класа - класа

- Оно што су ентитети у ER дијаграму, то су класе у UML дијаграму класа.
- Класа се састоји од:
 - имена
 - атрибута
 - информације о кључу
- Класа не мора да садржи (прецизне) типове атрибута, јер конструишемо концептуални модел који треба бити апстрактна слика домена који моделујемо.
- Методе класе се не наводе, моделујемо само структуру података.



UML дијаграм класа - пример класе

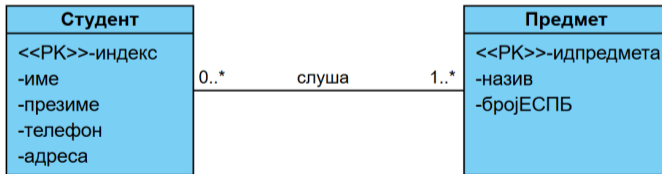
- **<<PK>>** представља *стереотип* помоћу којег означавамо кључне атрибуте. Уместо тога, кључни атрибут се може означити сличицом кључа поред атрибута на који се односи.





UML дијаграм класа - асоцијација

- *Асоцијација* представља именовани однос између два или више објеката.
- На пример: Студент и Предмет улазе у однос Слуша - Студент слуша предмете и предмет слушају студенти.



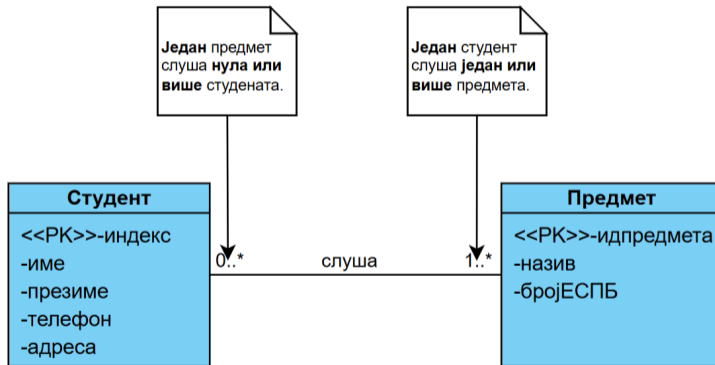


UML дијаграм класа - кардиналност односа

- Кардиналност односа (смер Класа А -> Класа Б) дефинише најмањи могући m и највећи могући n број појављивања објекта класе Б за једно појављивање објекта класе А. Кардиналност $m..n$ се у овом случају наводи уз класу Б.
- Уколико се кардиналност не наведе, подразумевано је $1..1$, или само 1 скраћено.
- Кардиналност $m..*$ значи да не постоји горње ограничење.
- Кардиналност $0..*$ се може скраћено означити само са $*$.



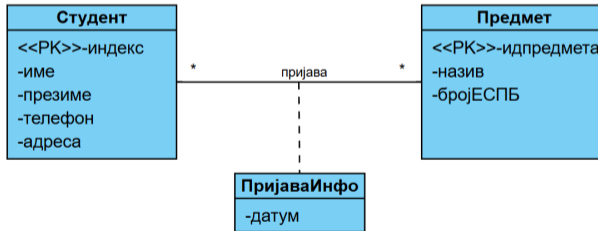
Пример - кардиналност односа студент-предмет





UML дијаграм класа - класа асоцијације

- *Класа асоцијације* - однос између објеката класа који садржи атрибуте.
- На пример: Студент пријављује предмете које жели да полаже, и за сваку пријаву желимо да сачувамо датум пријаве.





Пример - везе између објеката исте класе

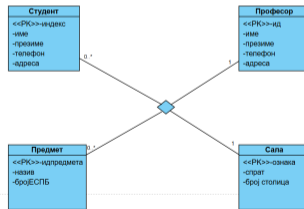
- Објекти исте класе могу међусобно улазити у однос, формирајући *цикличне односе*.
- На пример: Студент може менторисати најмање 10 а највише 20 студената, док једног студента може менторисати тачно један студент.





UML дијаграм класа - n -арни односи

- UML дијаграми класа омогућавају моделовање односа са више од 2 ентитета. У том случају је потребно нацртати ромб с којим се повезују све класе које улазе у тај однос.





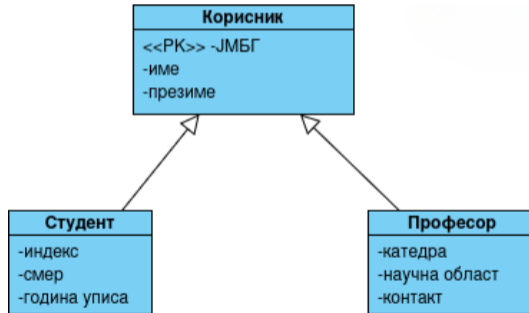
UML дијаграм класа - наслеђивање

- Успостављање хијерархије међу класама
- (Базна) класа улази у однос са класама које су изведене из ње.
- Базна класа садржи атрибуте који су заједнички са све изведене класе и као таква представља **генерализацију** изведених класа.
- Изведена класа садржи атрибуте на основу којих се разликује од осталих изведених класа и као таква представља **специјализацију** базне класе.



UML дијаграм класа - наслеђивање

- Однос између базне и изведене класе се успоставља цртањем непопуњеног троугла у смеру ка базној класи.





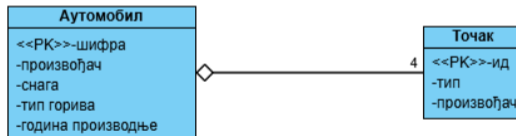
UML дијаграм класа - агрегација

- *Агрегација* представља тип односа у којем објекат једне класе представља део објекта неке друге класе.
- Делови могу постојати независно од целине (имају свој кључ).
- На пример: точак представља део аутомобила, али може постојати и независно од њега.
- Овај тип односа се може моделирати и асоцијацијом, али то није у свим случајевима семантички исправно.



UML дијаграм класа - агрегација

- Веза се успоставља цртањем непопуњеног ромба у смеру ка класи која представља целину.





UML дијаграм класа - композиција

- *Композиција* такође представља тип односа у којем објекат једне класе представља део објекта друге класе.
- Објекти који представљају делове **не могу** постојати независно од објекта који представља целину (не морају имати свој кључ).
- На пример ставка рачуна чини део рачуна (целина) и не може постојати ван њега.



UML дијаграм класа - композиција

- Веза се успоставља цртањем попуњеног ромба у смеру ка класи која представља целину.





Садржај

- 1 UML дијаграм класа
- 2 **Пример - издавачко предузеће**
- 3 Пример - организовање музичких концерата
- 4 Додатак



Пример - издавачко предузеће

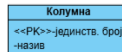
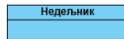
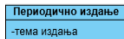
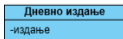
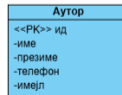
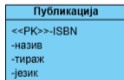
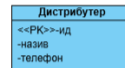
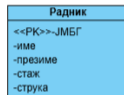
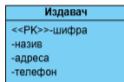
- Потребно је моделовати систем за једно издавачко предузеће. У бази је потребно чувати информације о раднику: ЈМБГ, име, презиме, стаж и струка. Радник може бити запослен код неког издавача (највише једног), а такође може возити и за неког дистрибутера (највише једног). Издавач и дистрибутер може бити у односу са више радника.
- За издавача се чувају шифра, назив, адреса и телефон док се за дистрибутера чувају идентификатор дистрибутера, назив и телефон.
- Издавач објављује публикације, за које се чувају ISBN, назив, тираж и језик. С друге стране, издавач мора објавити барем једном неку публикацију, а може и више пута. Једну публикацију мора објавити барем један издавач.



- Публикација може бити дневна новина, у том случају се чува и издање, недељник или периодично издање, у том случају се чува тема издања.
- Недељник садржи барем једну колумну. Колумна се налази у тачно једном недељнику и одликује се јединственим бројем и називом.
- За сваког аутора чува се идентификатор, име, презиме, телефон и имејл. Аутор може написати неколико колумни, док једну колумну може написати само један аутор.

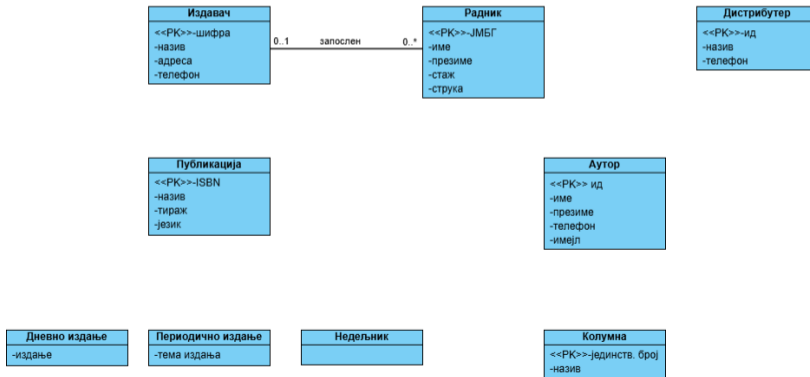


Корак 1. Цртање класа и њихових атрибута



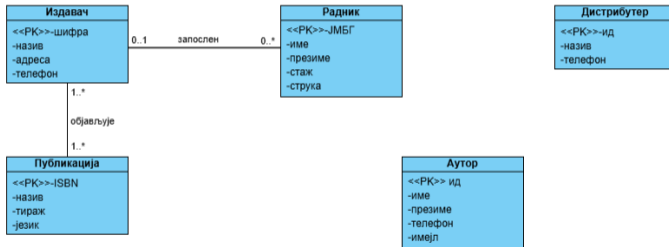


Корак 2. Однос између издавача и радника





Корак 3. Однос између издавача и публикације



Дневно издање
 -издање

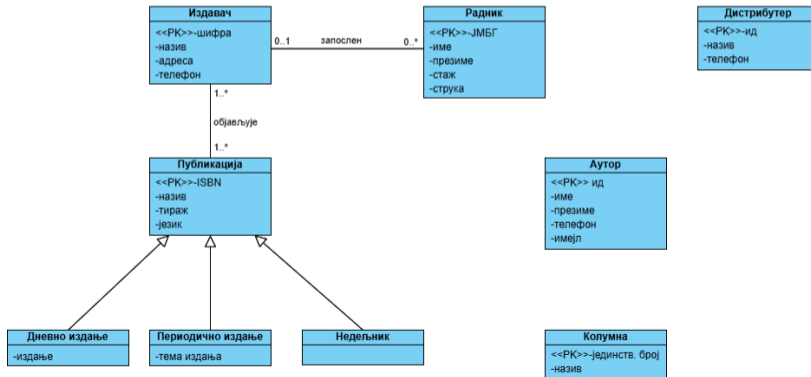
Периодично издање
 -тема издања

Недељник

Колумна
 <<РК>>-јединств. број
 -назив



Корак 4. Додавање специјализације класе *Публикација*

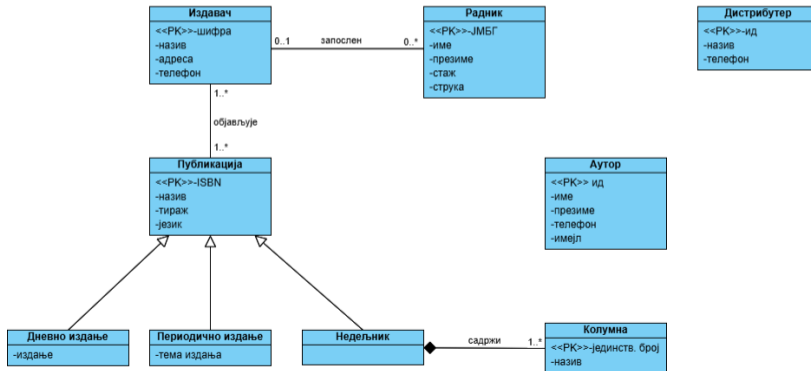




Да ли би имало смисла додати још негде специјализацију/генерализацију?



Корак 5. Додавање композиције Колумна-Недељник



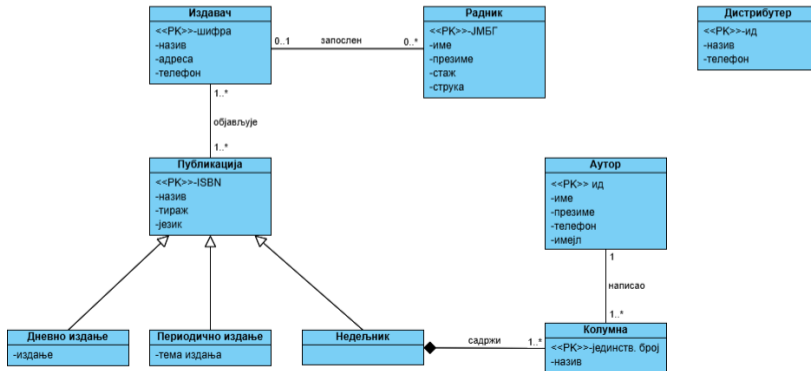


Зашто је однос између колумне и недељника моделован композицијом, а не асоцијацијом или агрегацијом?

- Уколико би веза била моделована асоцијацијом или агрегацијом, уништењем објекта класе недељник, сви објекти класе Колумна би наставили да постоје, што би било погрешно. Колумна постоји докле год постоји недељник чији је она део. Из тог разлога се тај однос моделира композицијом.

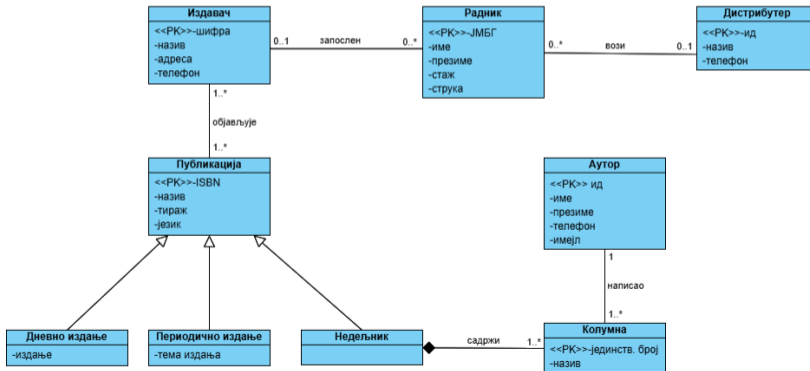


Корак 6. Однос између аутора и колумне





Корак 7. Однос између дистрибутера и радника





Садржај

- 1 UML дијаграм класа
- 2 Пример - издавачко предузеће
- 3 **Пример - организовање музичких концерата**
- 4 Додатак



Пример - организовање музичких концерата

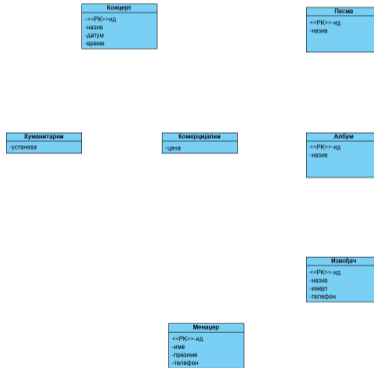
- Потребно је моделовати систем за организовање музичких концерата. У бази је потребно чувати информације о концертима (идентификатор, назив, датум и време). Концерт може бити хуманитарни или комерцијални.
- Хуманитарни концерт треба да садржи информацију о установи на коју се односи, док комерцијални концерт треба да садржи информацију о цени карте. Комерцијални концерт уговара менаџер са извођачем са којим има склопљен уговор. Потребно је чувати информацију о броју уговора и проценту зараде који узима менаџер. Један менаџер може уговорити концерт на којем ће бити више извођача (најмање један).
- За сваког менаџера у бази се чувају: идентификатор, име, презиме, број телефона. За сваког извођача у бази се чувају: идентификатор, назив, телефон, имејл.



- Извођач може снимити више албума, а не мора ни један. За сваки албум се чувају идентификатор и назив. За сваки албум мора постојати барем један извођач у бази.
- Албум садржи једну или више песама. Свака песма се карактерише идентификационим бројем и називом. Песма не мора бити на неком албуму, али ако јесте, онда је највише на једном. Једну песму може изводити више извођача, али мора барем један.
- Извођач може свирати на концертима, у ком случају се за сваки концерт чува термин у коме свира. Приликом свирања на концерту изводи једну или више песама. Са друге стране, у бази се могу налазити песме које нису ни једном биле изведене на концерту.

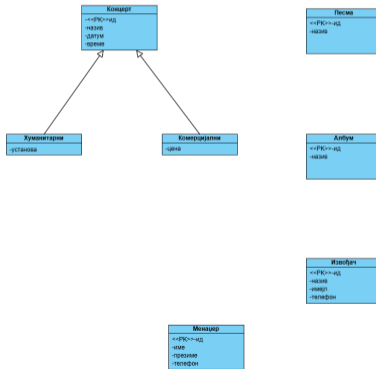


Корак 1. Цртање класа и њихових атрибута



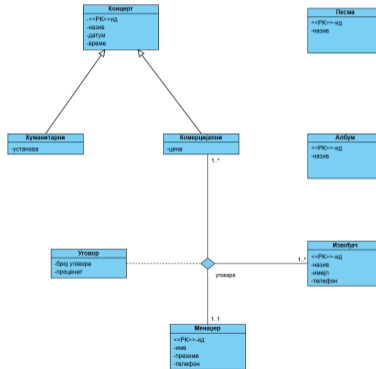


Корак 2. Додавање специјализације класе *Концерт*





Корак 3. Однос *Комерцијални-Извођач-Менаџер* и додавање класе асоцијације





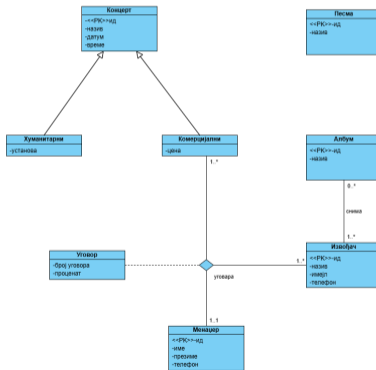
Тумачење кардиналности у тернарном односу

Претходни тернарни однос се посматра из 3 угла:

- **један пар** (извођач, комерцијални) улази у однос са **тачно једним** менаџером, обзиром да извођач може имати само једног менаџера, самим тим само један менаџер му може уговорити концерт; из тог разлога поред ентитета *Менаџер* се налази кардиналност **1..1**.
- **један пар** (менаџер, комерцијални) улази у однос са **1 или више** извођача, јер на једном концерту који је уговорио менаџер, може наступати више извођача; из тог разлога се поред ентитета *Извођач* налази кардиналност **1..***.
- Аналогно за однос између (менаџер, извођач) и комерцијални.

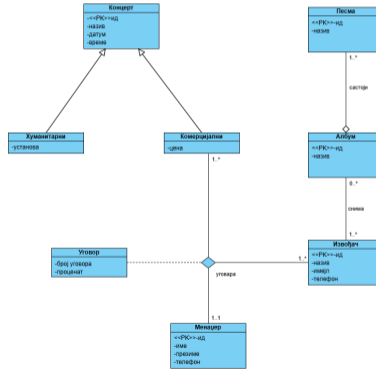


Корак 4. Однос између извођача и албума





Корак 5. Агрегација између албума (целина) и песме (део)

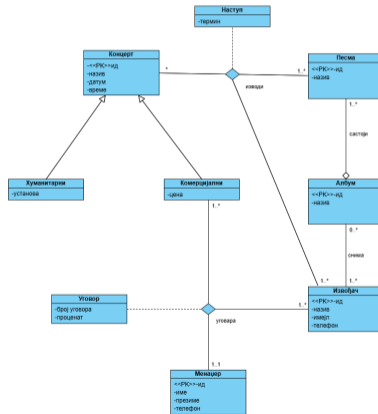




Претходни однос је могао бити моделован и *асоцијацијом*, али уколико желимо да нагласимо да је песма саставни део албума и да може наставити да постоји и након што албум престане да постоји, онда је семантички исправније користити *агрегацију*.



Корак 6. Однос *Извођач-Концерт-Песма* и додавање класе асоцијације





Садржај

- 1 UML дијаграм класа
- 2 Пример - издавачко предузеће
- 3 Пример - организовање музичких концерата
- 4 **Додатак**



Додатак

- За цртање дијаграма класа података може се користити *Visual Paradigm* алат: <https://online.visual-paradigm.com/diagrams/solutions/free-class-diagram-tool/>