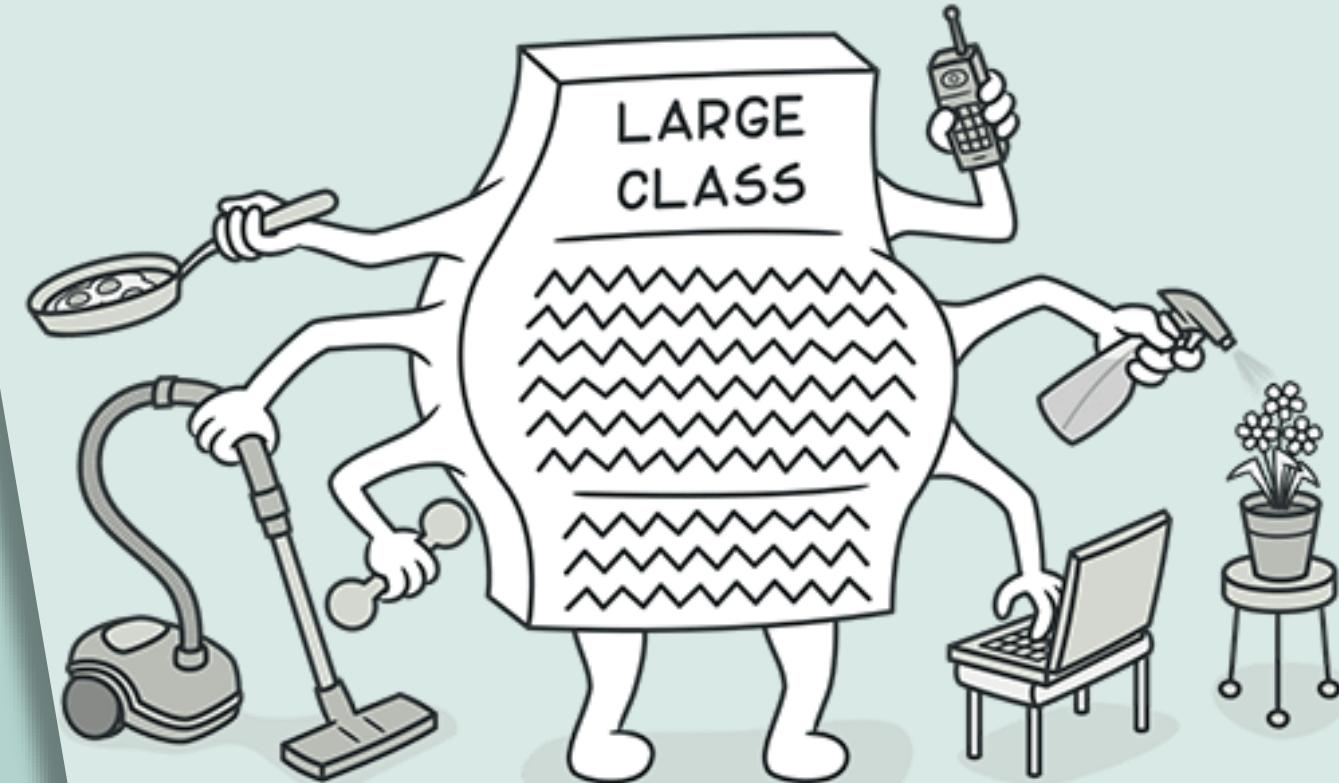


Uvod u softversko inženjerstvo

Čiste klase

Nikola Luburić
nikola.luburic@uns.ac.rs



Navedi teme koje misliš da treba **više** pokriti kroz predavanja

Ne znam koliko je dobra ideja da ovo bude deo predavanja, ali bih voleo da imamo više primera gde možemo da vežbamo primenu naučenih veština (pa makar i u formi materijala, uzimajući u obzir koliko malo vremena zapravo imamo na predavanju, iako nam termin traje tri sata).

Више приче о послу, како искусан инжењер размишља кад добије проблем (одакле крене да рефакторише, процес размишљања итд).

Primena na projektu

Волео бих да чујем више где сте ви као предавач грешили у послу и током студија. На грешкама се много учи.

Čiste klase

Greške u inženjerstvu softvera

- ✓ 5. Konstruiši klase koje imaju jednu odgovornost

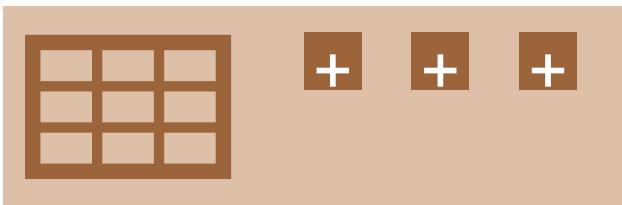
✓ 1. Definiši jednu ulogu koju klasa ima u aplikaciji

✓ 2. Povećaj strukturalnu koheziju klase

Koje su uloge?

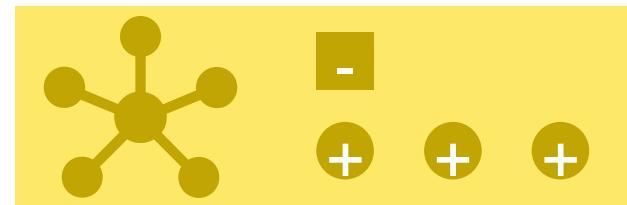
✓ 3. Povećaj semantičku koheziju klase

✓ 4. Redukuj spregnutost klase sa drugim klasama



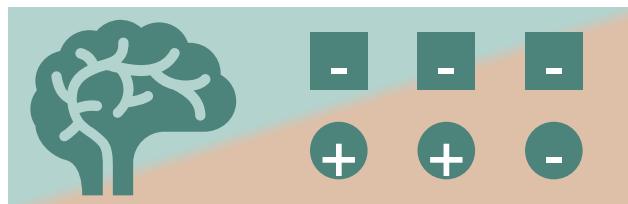
Struktura podataka

- Izlaže stanje
- Ne sadrži metode



Koordinatori

- Ne poseduje promenljivo stanje
- Pamet metoda je u koordinaciji



Pametna klasa

- Sakriva stanje
- Pamet metoda je u **poslovnoj ili tehničkoj logici**



Pametna funkcionalna klasa

- Ne poseduje promenljivo stanje
- Pamet metoda je u **poslovnoj ili tehničkoj logici**

✓ 5. Konstruiši klase koje imaju jednu odgovornost

✓ 1. Definiši jednu ulogu koju klasa ima u aplikaciji

✓ 2. Povećaj strukturalnu koheziju klase

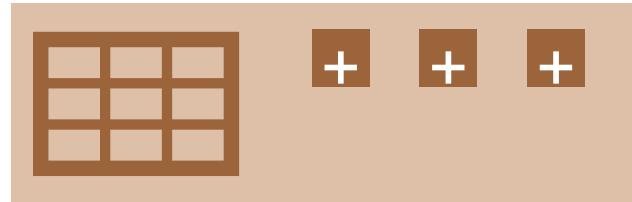


✓ 3. Povećaj semantičku koheziju klase



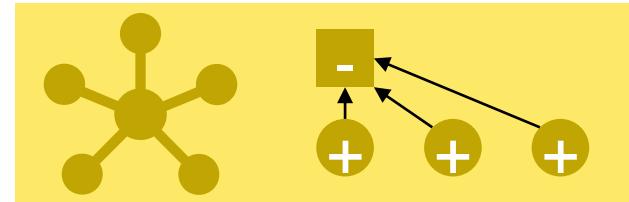
✓ 4. Redukuj spregnutost klase sa drugim klasama





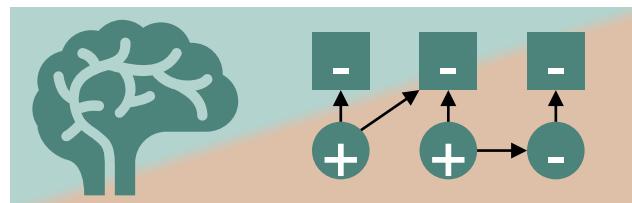
Struktura podataka

- Nema strukturalnu koheziju



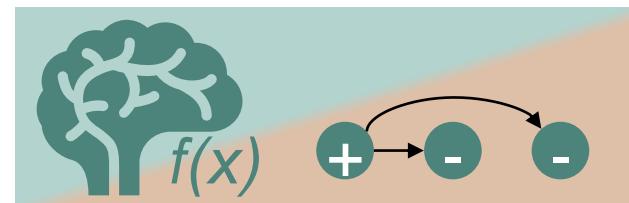
Koordinatori

- Treba da ima visoku strukturalnu koheziju



Pametna klasa

- Treba da ima visoku strukturalnu koheziju



Pametna funkcionalna klasa

- Treba da ima visoku strukturalnu koheziju između metoda

- ✓ 5. Konstruiši klase koje imaju jednu odgovornost

✓ 1. Definiši jednu ulogu koju klasa ima u aplikaciji

✓ 2. Povećaj strukturalnu koheziju klase

✓ 3. Povećaj semantičku koheziju klase

✓ 4. Redukuj spregnutost klase sa drugim klasama

Sve klase treba da imaju visoku semantičku koheziju

User vs Person

✓ 5. Konstruiši klase koje imaju jednu odgovornost

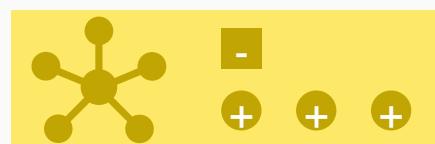
✓ 1. Definiši jednu ulogu koju klasa ima u aplikaciji

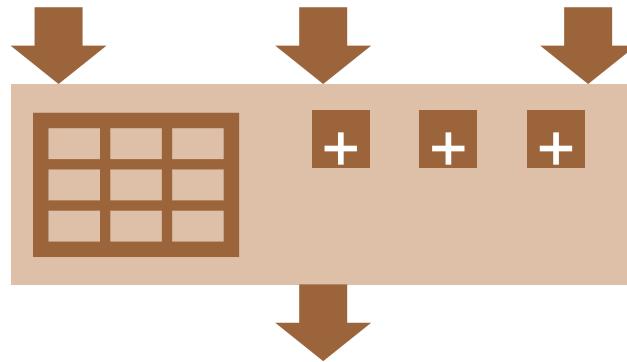
✓ 2. Povećaj strukturalnu koheziju klase

✓ 3. Povećaj semantičku koheziju

✓ 4. Redukuj spregnutost klase sa drugim klasama

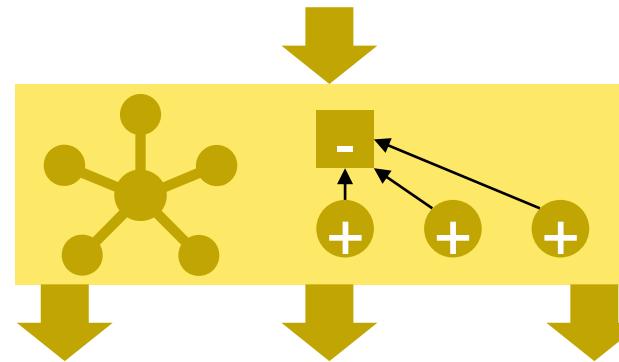
Šta su primeri sprege?





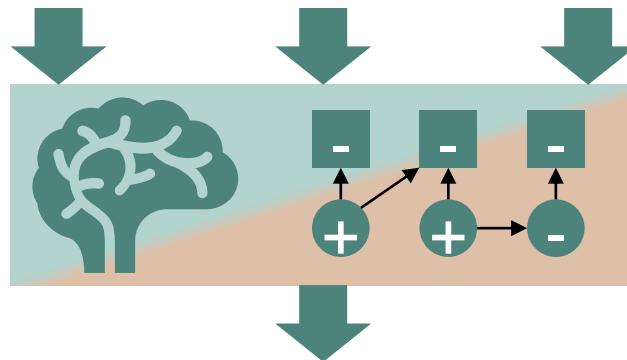
Struktura podataka

- Druge klase je referenciraju
- Slabo referencira druge klase



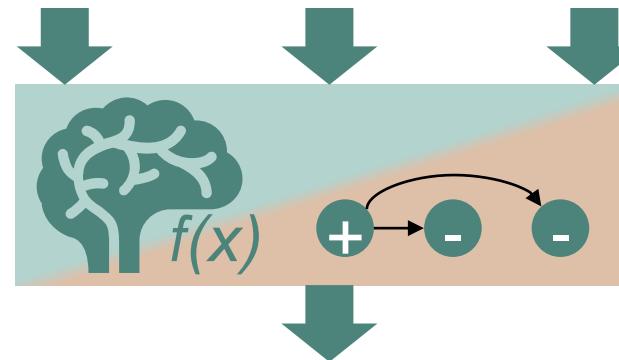
Koordinatori

- Druge klase je slabo referenciraju
- Referencira druge klase



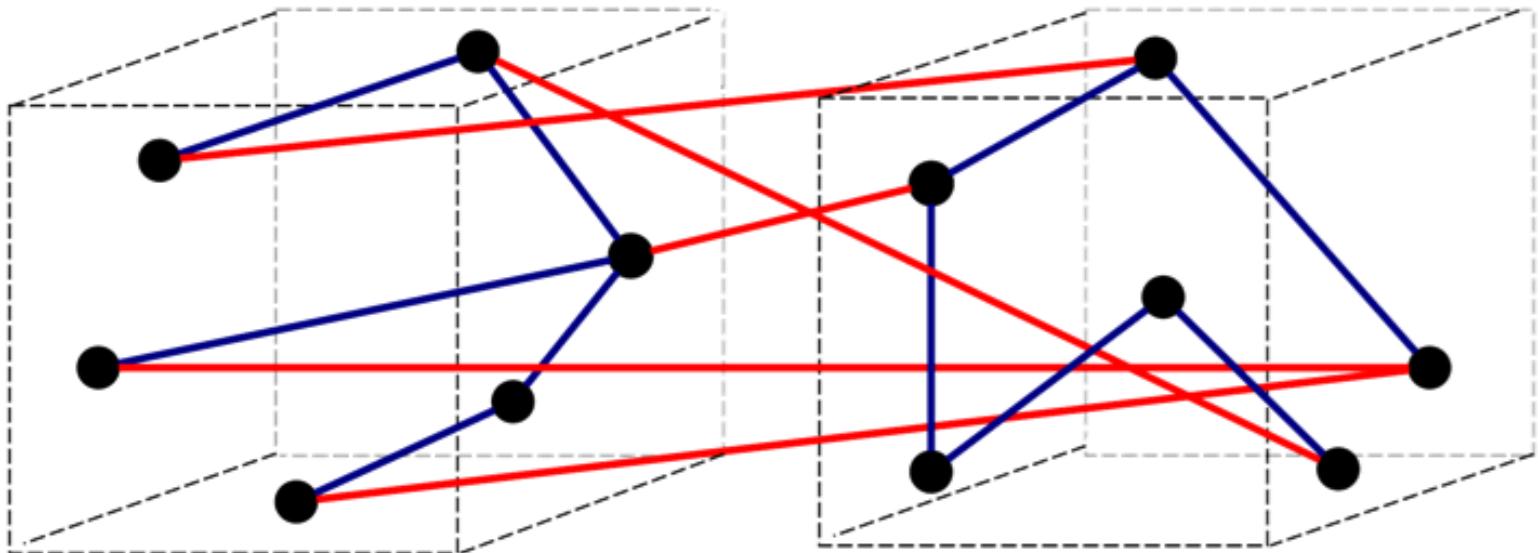
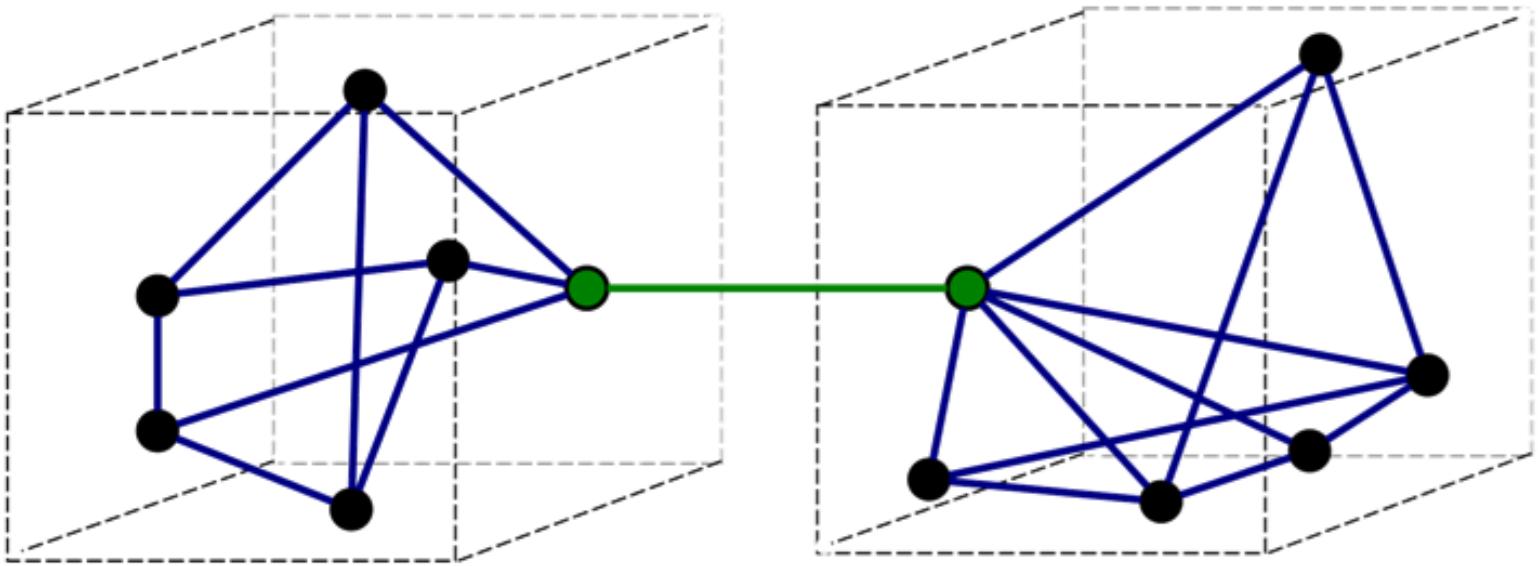
Pametna klasa

- Druge klase je referenciraju
- Slabo referencira druge klase



Pametna funkcionalna klasa

- Druge klase je referenciraju
- Slabo referencira druge klase





5. Konstruiši klase koje imaju jednu odgovornost



1. Definiši jednu ulogu koju klasa ima u aplikaciji

*1. Jasno definisana
ne-hibridna uloga*



2. Povećaj strukturalnu koheziju klase

*2a. Strukt. kohezija
koja odgovara ulozi*



3. Povećaj semantičku koheziju klase

*3. Visoka
semantička kohezija*



4. Redukuj spregnutost klase sa drugim klasama

*2b. Spregnutost koja
odgovara ulozi*

SRP

OCP

LSP

ISP

DIP

Skupi stvari koje se menjaju iz istih razloga

Razdvoj one koje se menjaju iz različitih

Employee

- + calculatePay() : double
- + save() : void
- + reportEmployee() : String
- + findById(int id) : Employee

šta menja nova baza?

šta menja nov sistem bonusa?

šta menja nov izveštaj?

SRP

OCP

LSP

ISP

DIP

Skupi stvari koje se menjaju iz istih razloga Razdvoj one koje se menjaju iz različitih

Employee

EmployeeRepository

+ save(Employee e) : void
+ findById(int id) : Employee

šta menja nova baza?

PaycheckCalculator

+ calculatePay(Employee e) : double

šta menja nov sistem bonusa?

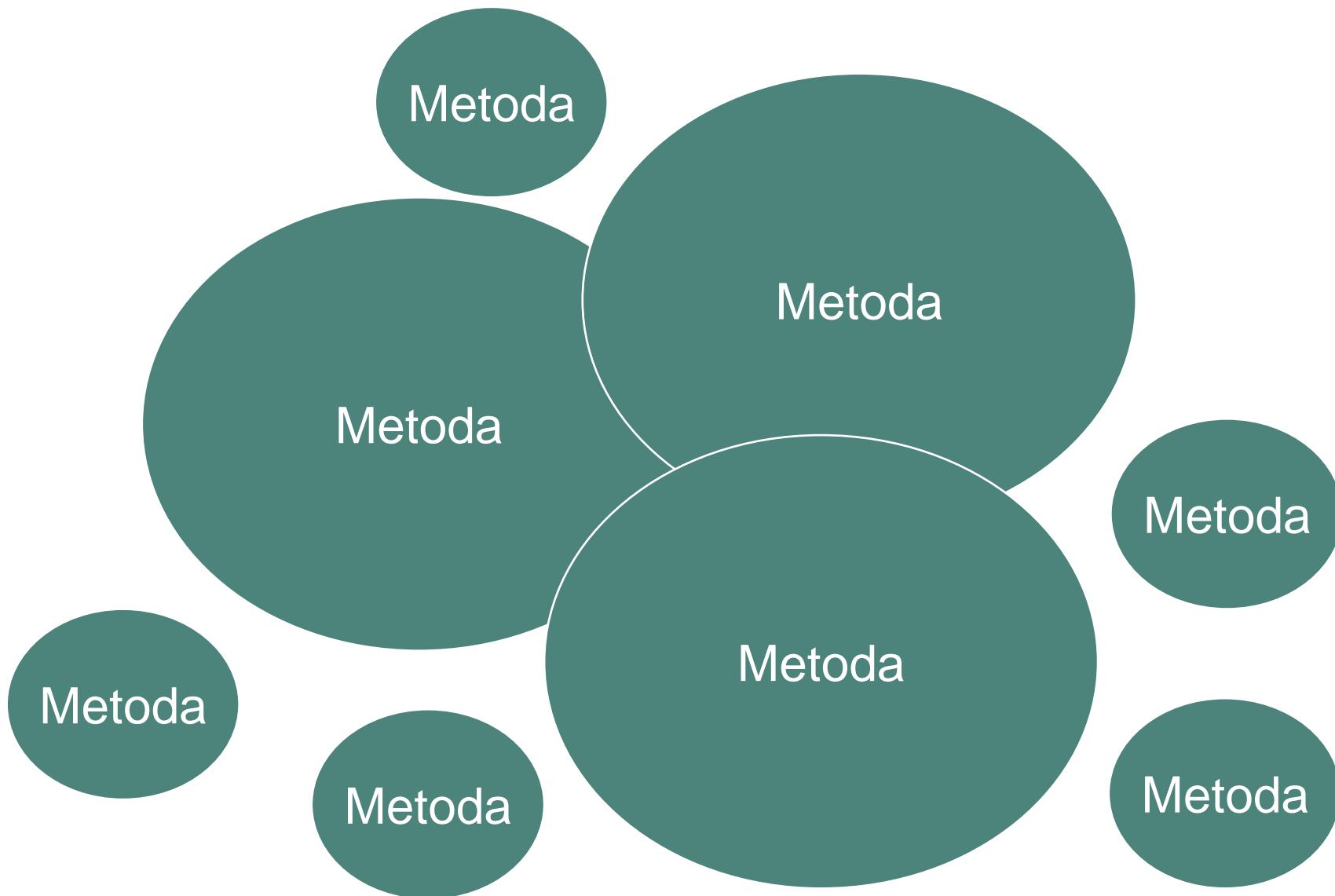
EmployeeReporter

+ report(Employee e) : String

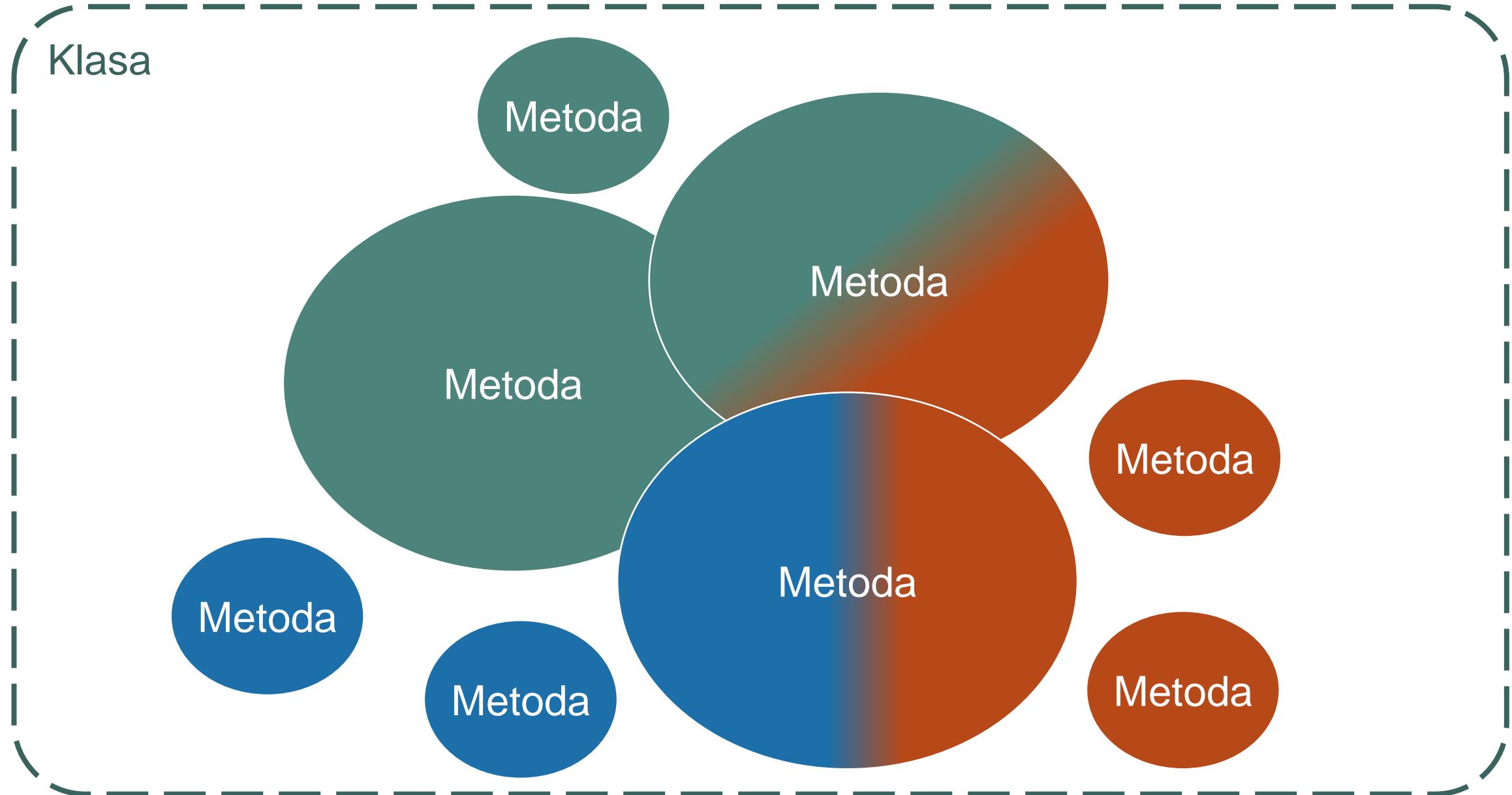
šta menja nov izveštaj?

Šta su posledice dobro napisanih metoda?

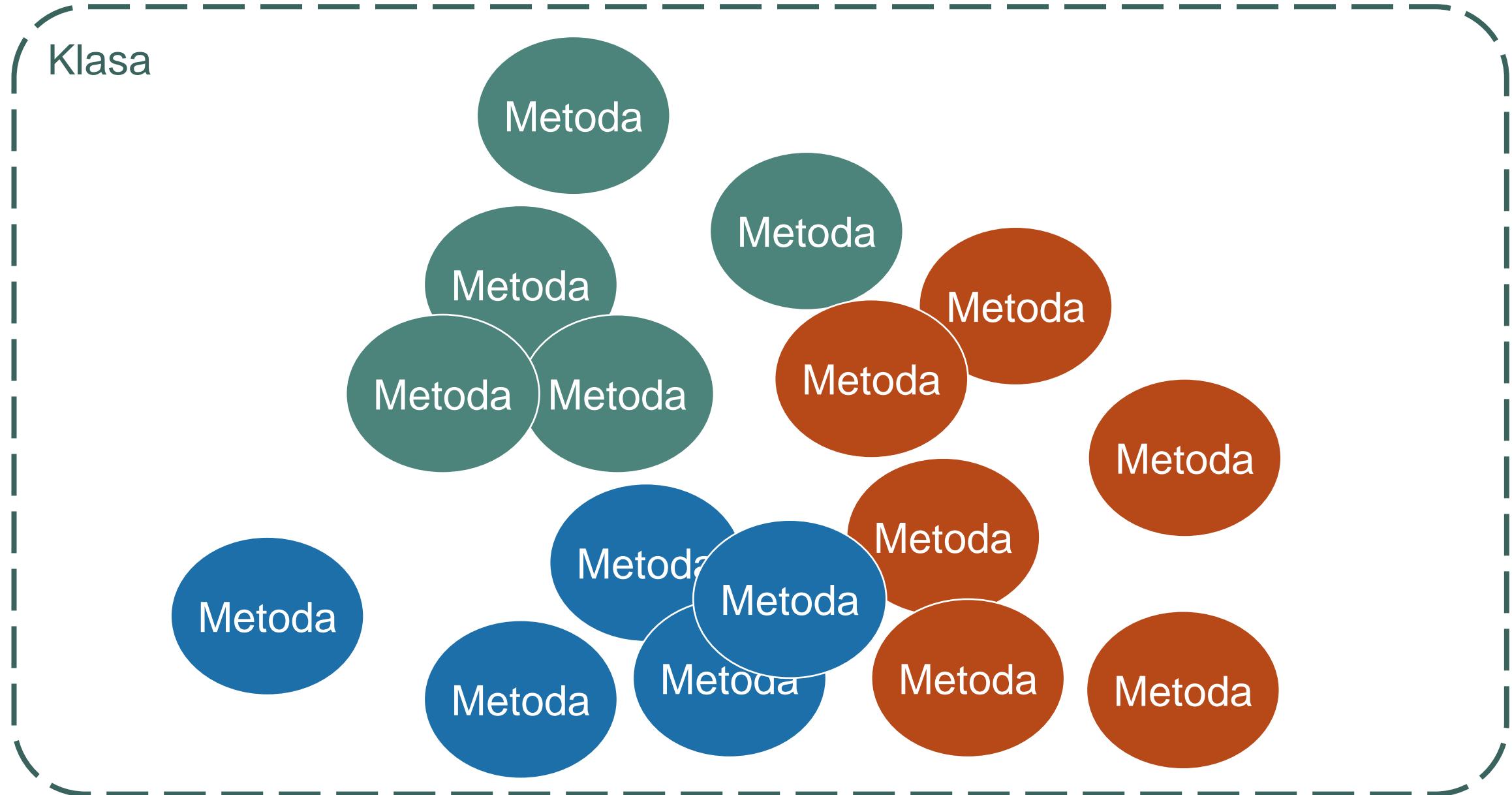
Klasa



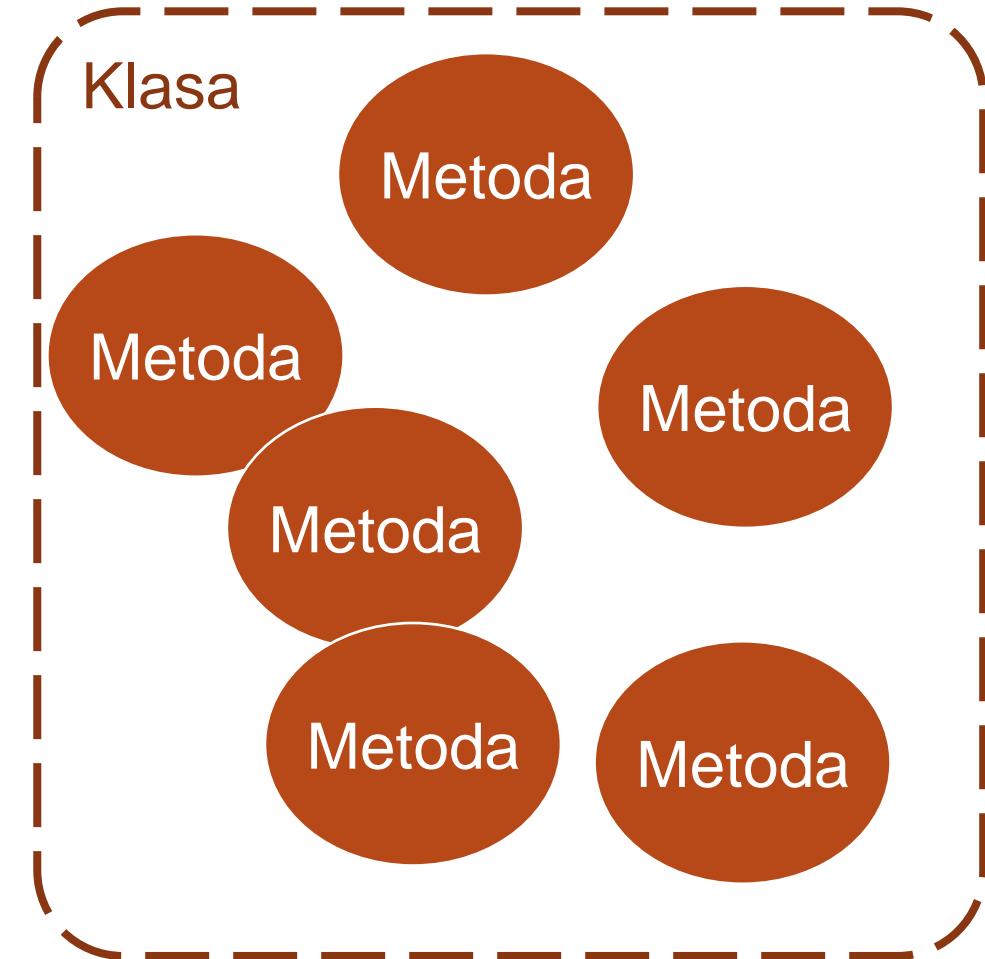
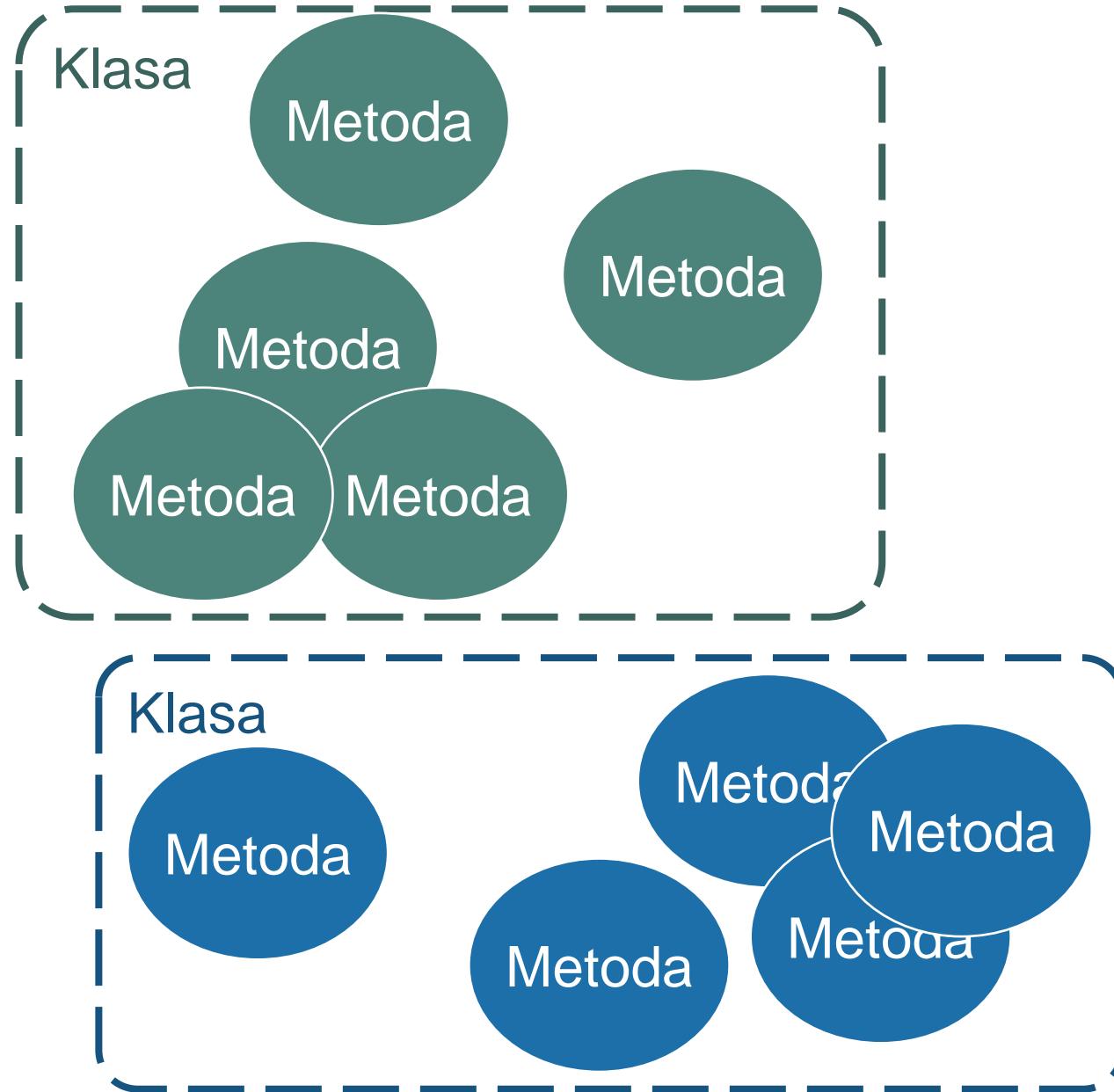
Šta su posledice dobro napisanih metoda?



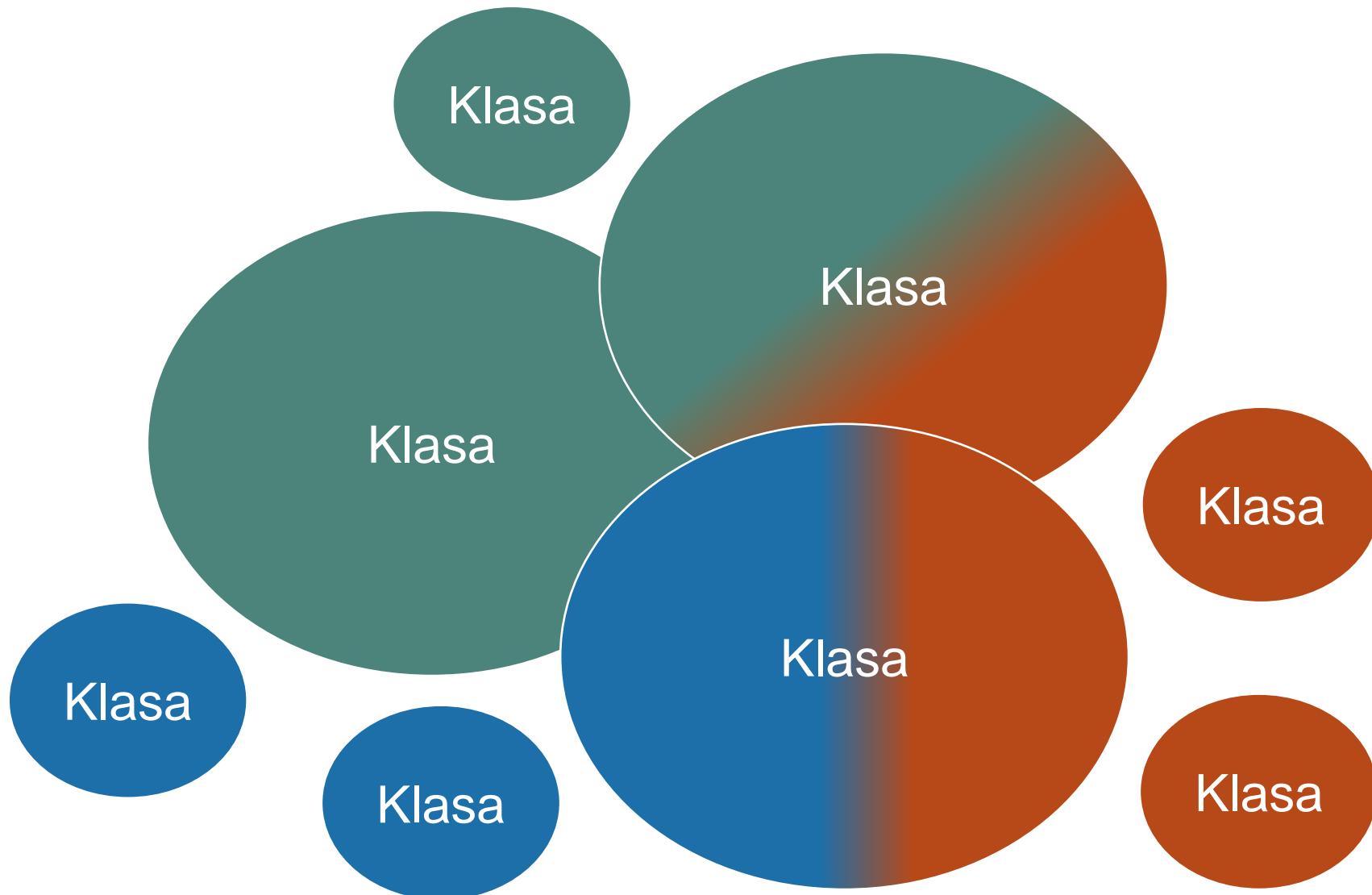
Šta su posledice dobro napisanih klasa?



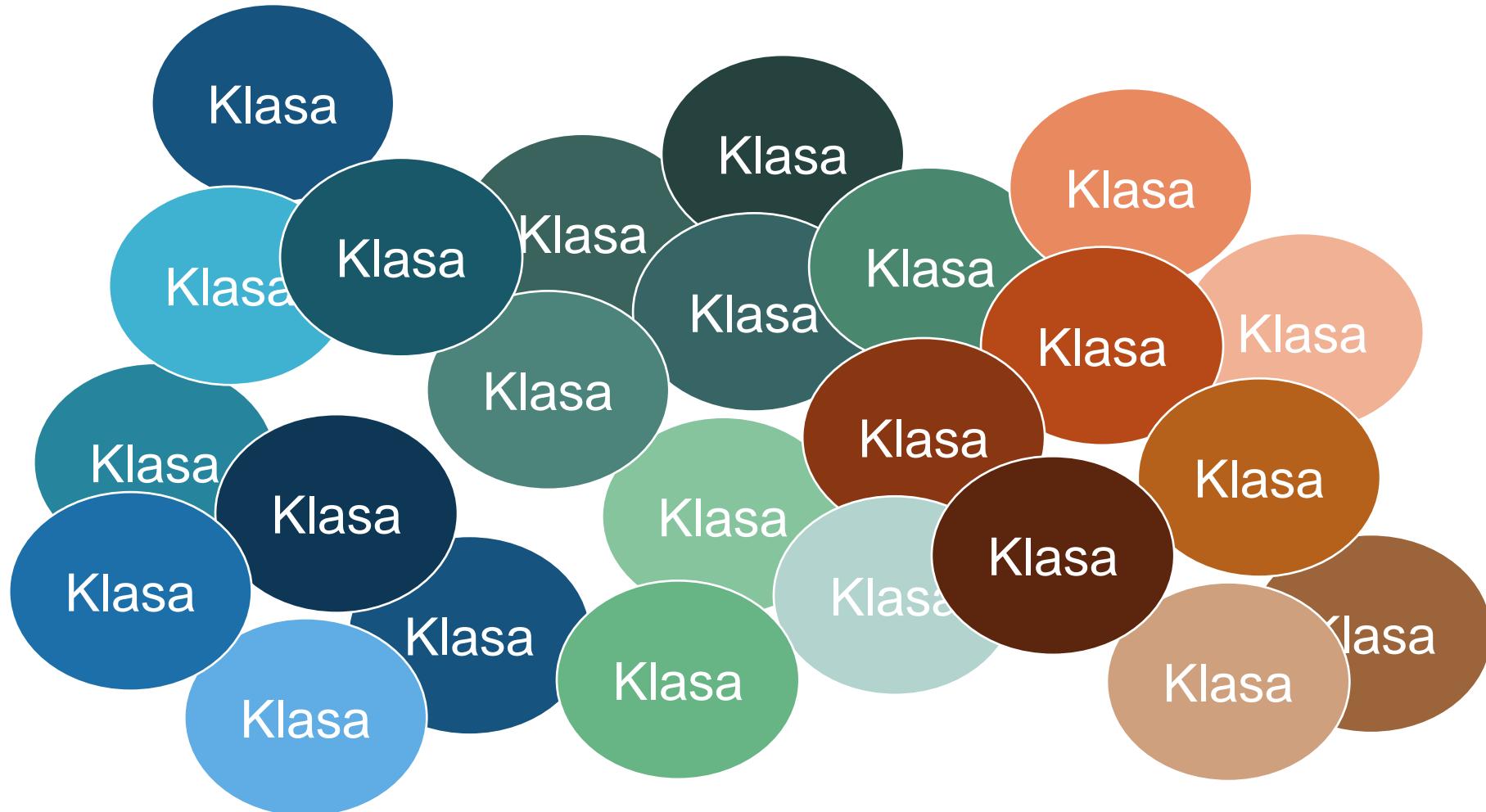
Šta su posledice dobro napisanih klasa?



Šta su posledice dobro napisanih klasa?

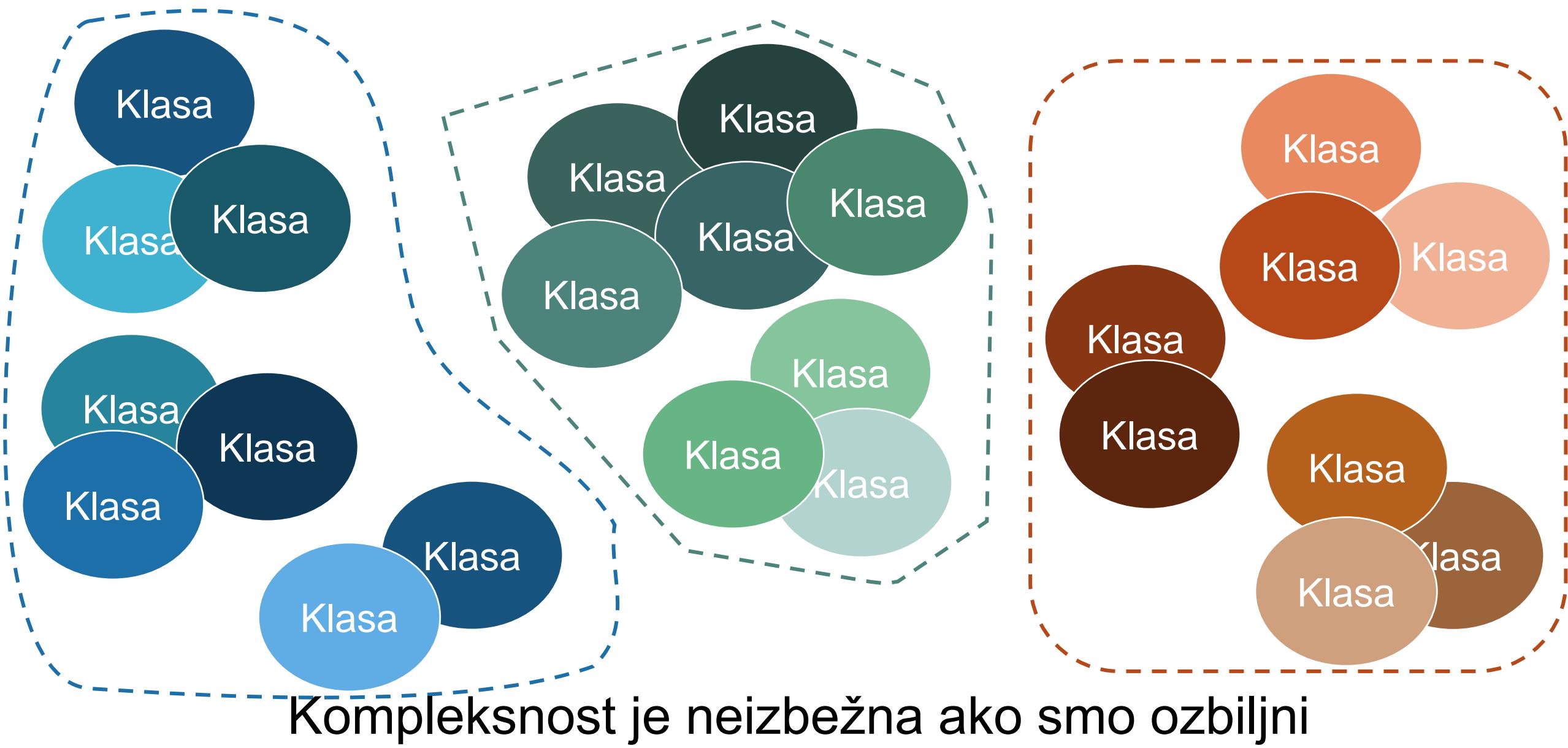


Šta su posledice dobro napisanih klasa?



Kompleksnost je neizbežna ako smo ozbiljni

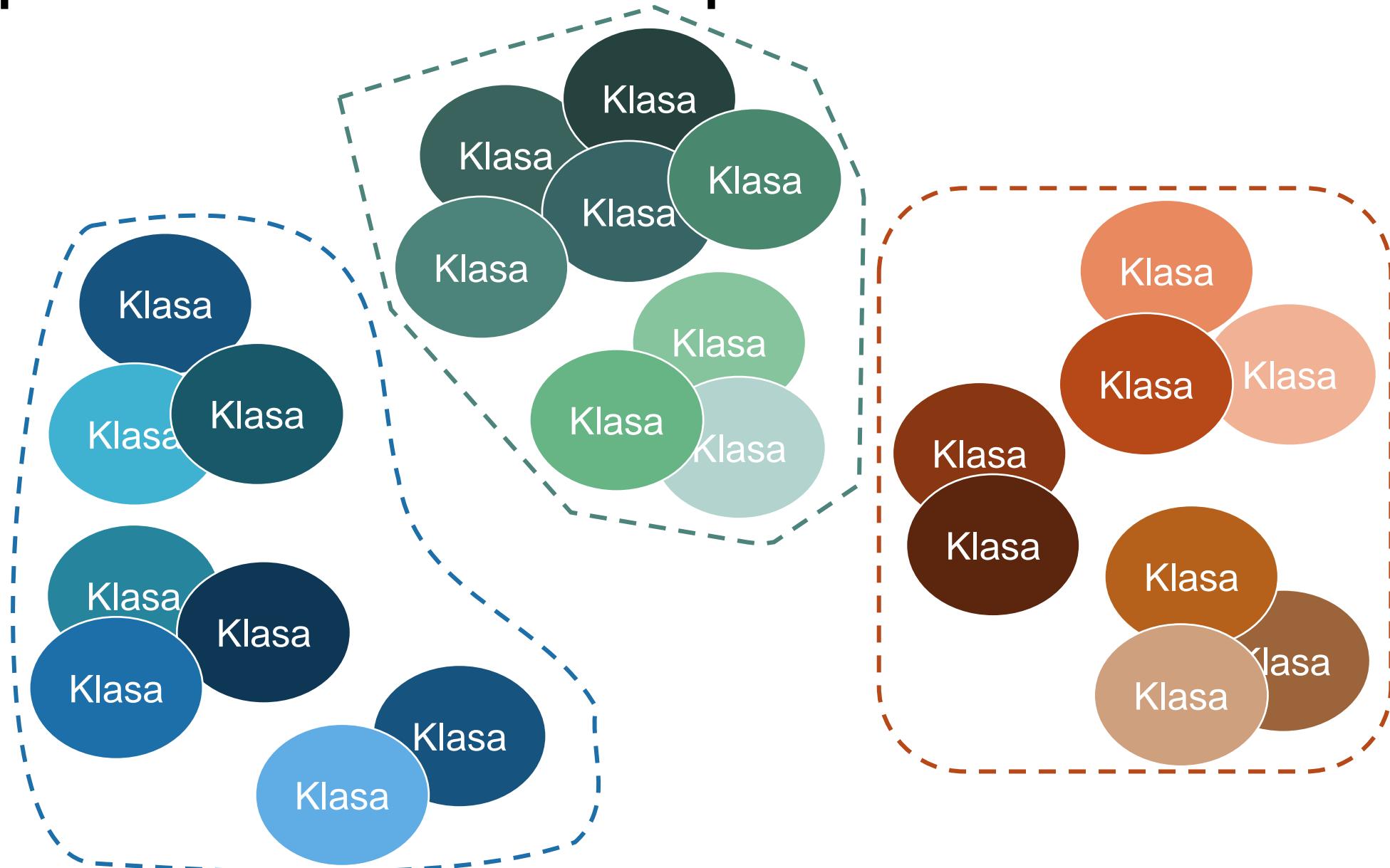
Šta su posledice dobro napisanih klasa?



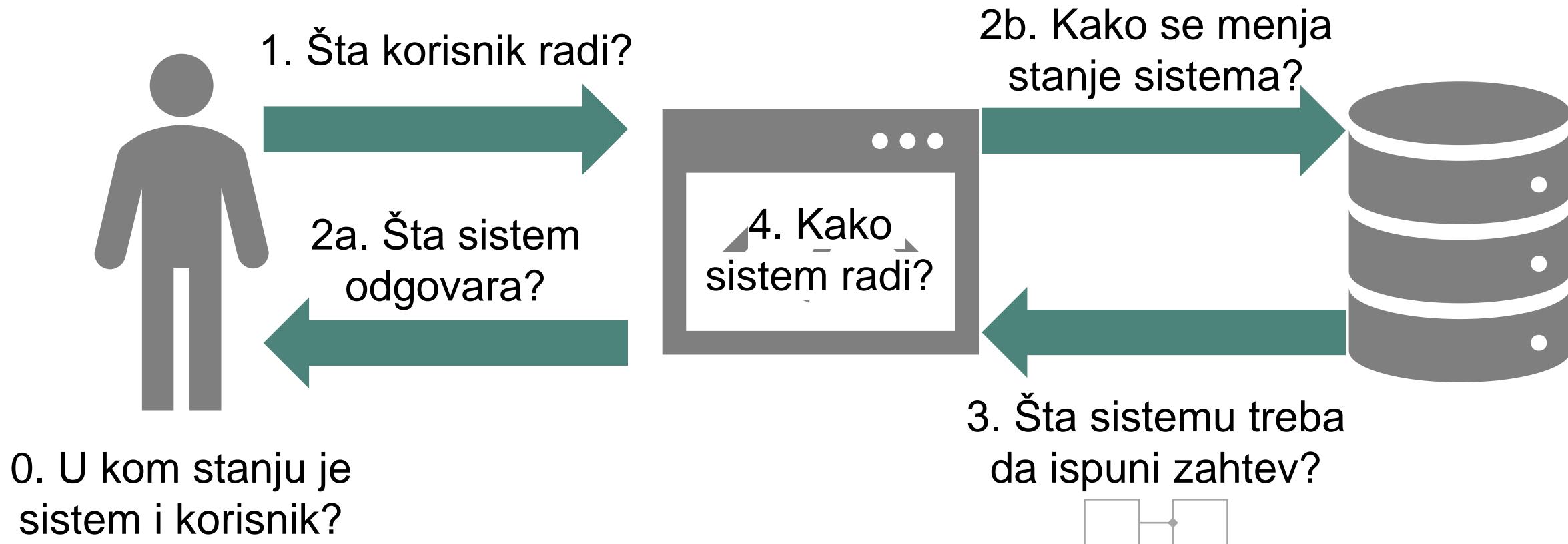
Šta su posledice dobro napisanih klasa?

Maintainability

- Modularity
- Reusability
- Analysability
- Modifiability
- Testability

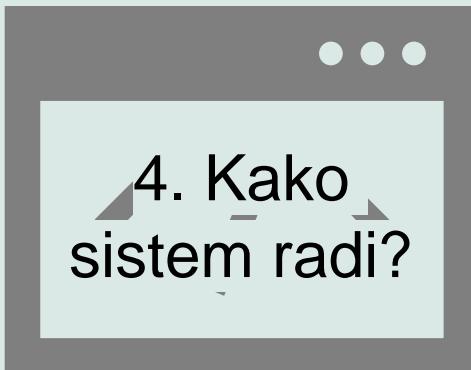


Predavač kreira termin za polaganje ispita i navodi strani jezik koji se polaže, nivo, maksimalan broj učenika, i datum polaganja ispita. Termini se ne mogu preklapati sa terminima drugih kurseva i ispita predavača, kao i sa terminima koji se održavaju uživo.



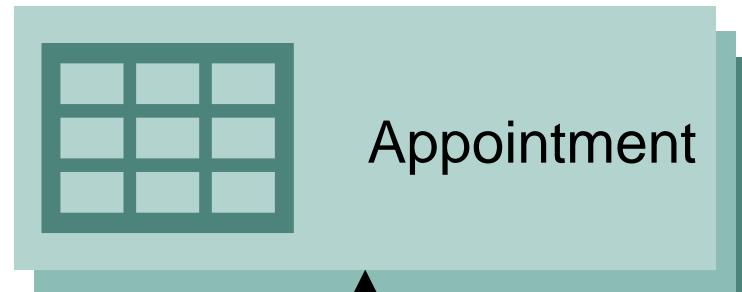
Predavač kreira termin za polaganje ispita i navodi strani jezik koji se polaže, nivo, maksimalan broj učenika, i datum polaganja ispita. Termini se ne mogu preklapati sa terminima drugih kurseva i ispita predavača, kao i sa terminima koji se održavaju uživo.

1. Šta korisnik radi?
- Jezik, nivo
 - # učenika
 - DateTime
 - *Predavač*



- Zauzeće predavača
- Zauzeće učionice

3. Šta sistemu treba da ispunji zahtev?



$$f(x)$$

Predavač kreira termin za polaganje ispita i navodi strani jezik koji se polaže, nivo, maksimalan broj učenika, i datum polaganja ispita. Termini se ne mogu preklapati sa terminima drugih kurseva i ispita predavača, kao i sa terminima koji se održavaju uživo.

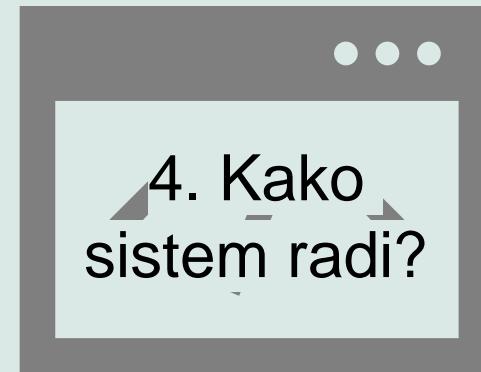
Direktor može da kreira termin ispita na identičan način kao i predavač, samo što će mu sistem odrediti koji predavač je najkompetentniji i u mogućnosti da u zadatom terminu održi ispit. Ako postoji, uzeti u obzir i prosečnu ocenu predavača.

Tip 76

Programmers Help People Understand What They Want

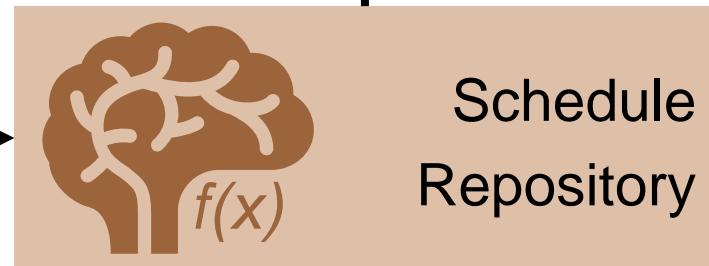
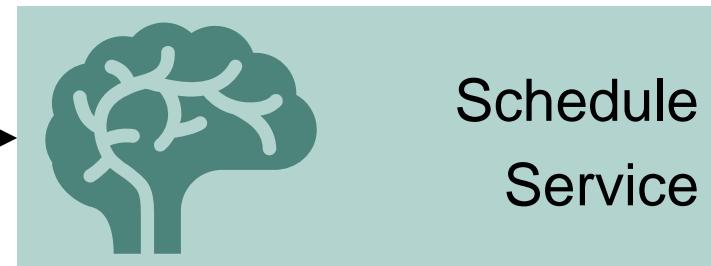
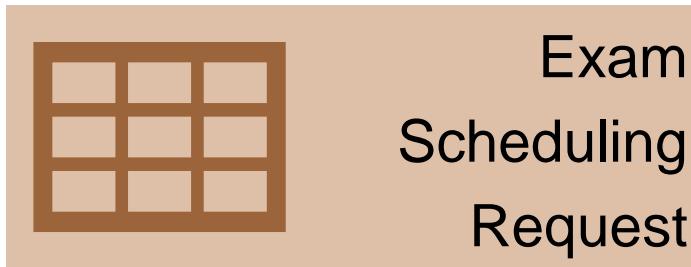
Direktor može da kreira termin ispita na identičan način kao i predavač, samo što će mu sistem odrediti koji predavač je najkompetentniji i u mogućnosti da u zadatom terminu održi ispit. Ako postoji, uzeti u obzir i prosečnu ocenu predavača.

1. Šta korisnik radi?
- Jezik, nivo
 - # učenika
 - DateTime
 - *Predavač*

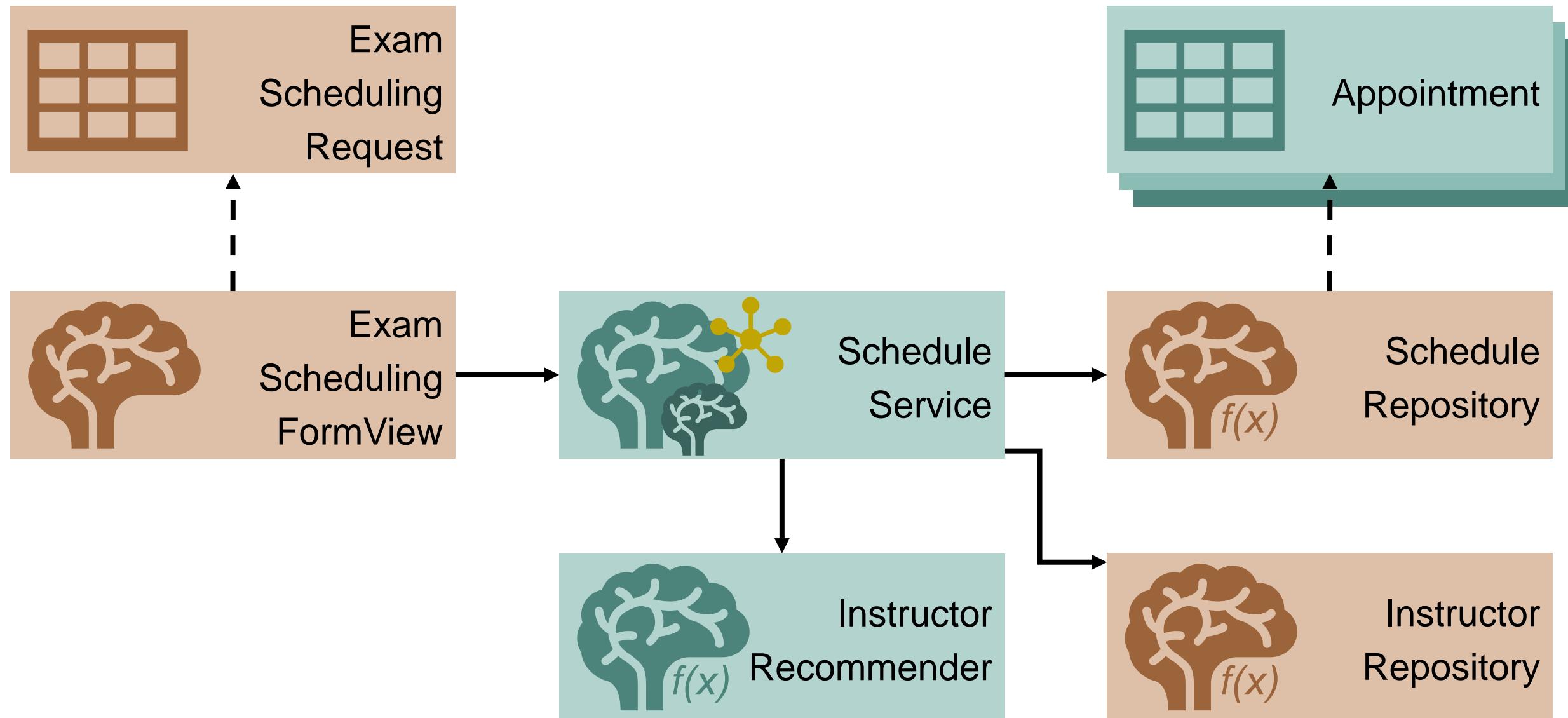


- Zauzeće predavača
- Zauzeće učionice

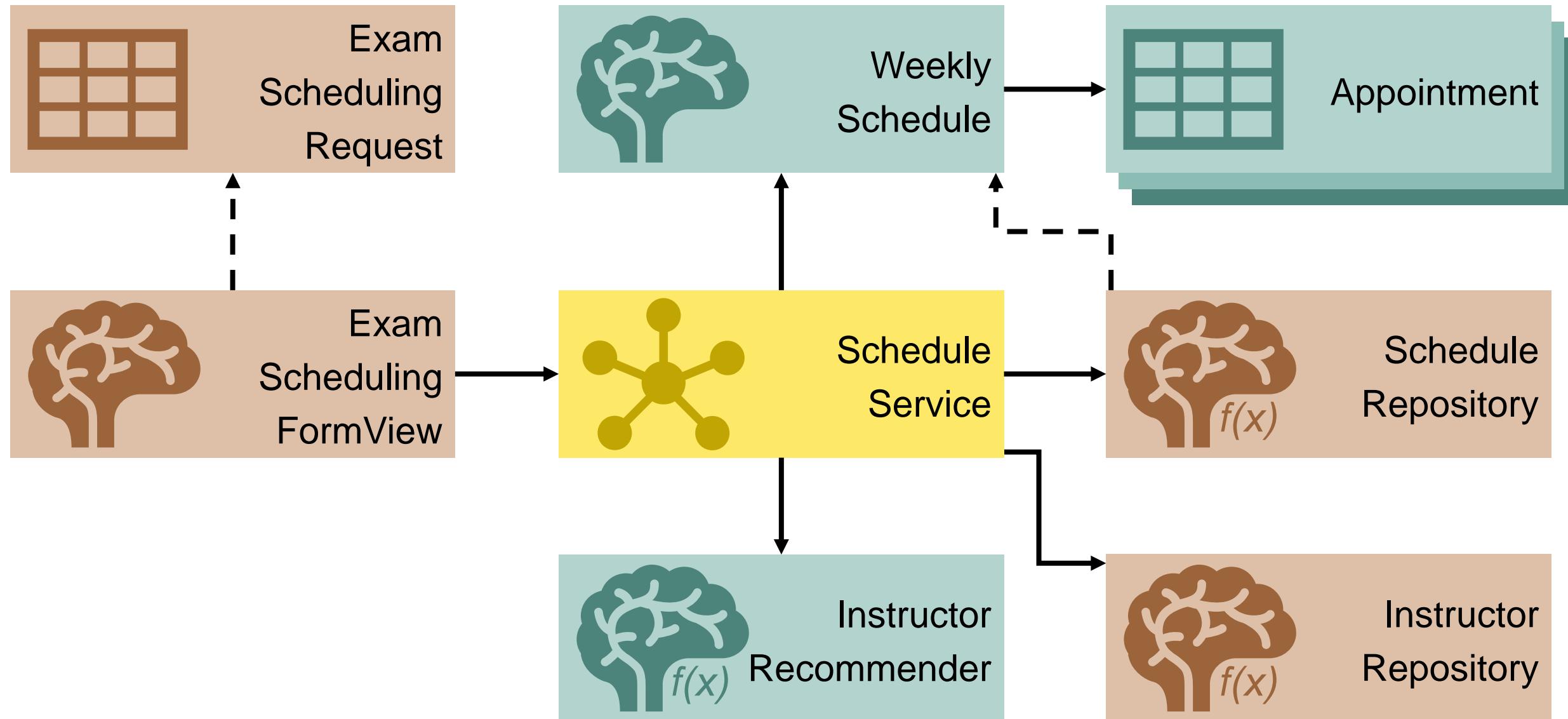
3. Šta sistemu treba da ispunji zahtev?



Direktor može da kreira termin ispita na identičan način kao i predavač, samo što će mu sistem odrediti koji predavač je najkompetentniji i u mogućnosti da u zadatom terminu održi ispit. Ako postoji, uzeti u obzir i prosečnu ocenu predavača.

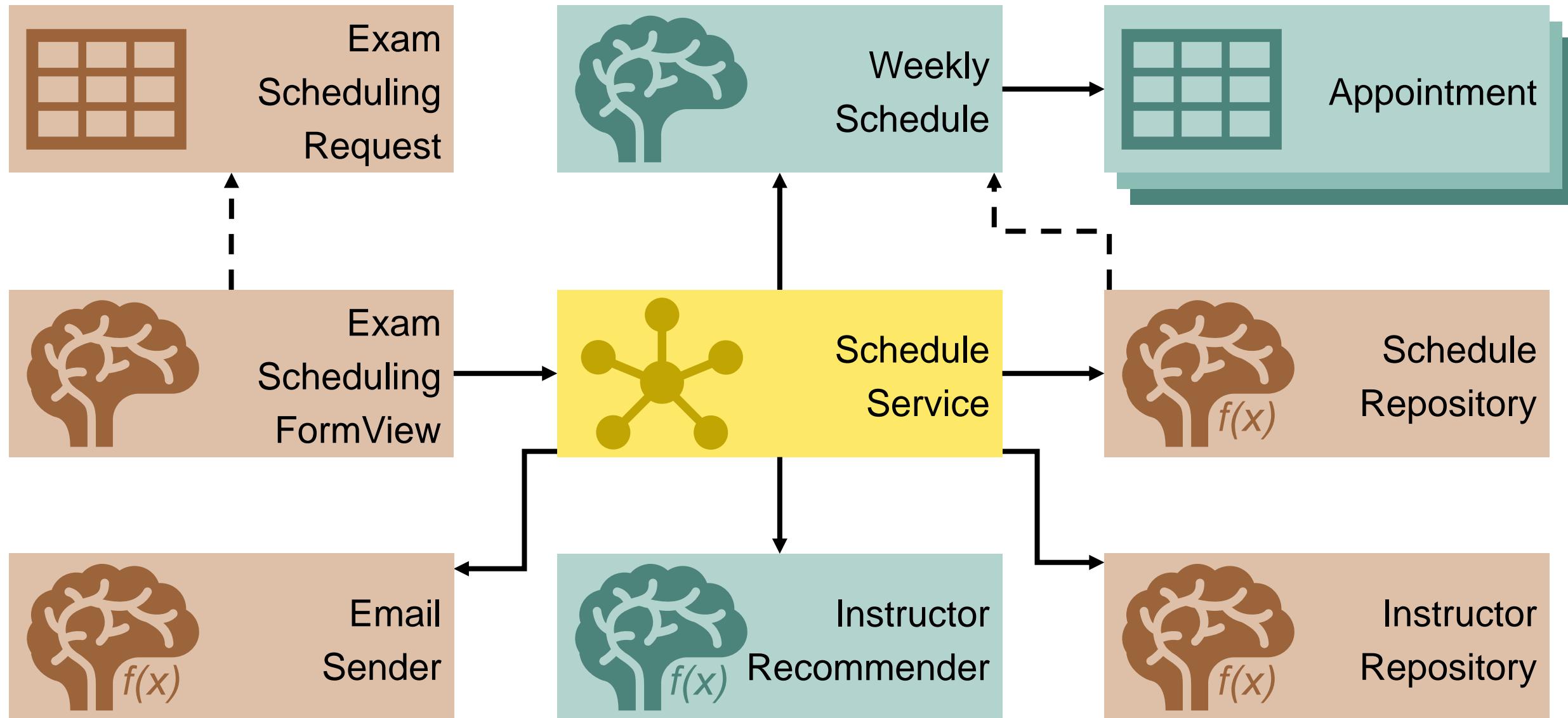


Direktor može da kreira termin ispita na identičan način kao i predavač, samo što će mu sistem odrediti koji predavač je najkompetentniji i u mogućnosti da u zadatom terminu održi ispit. Ako postoji, uzeti u obzir i prosečnu ocenu predavača.

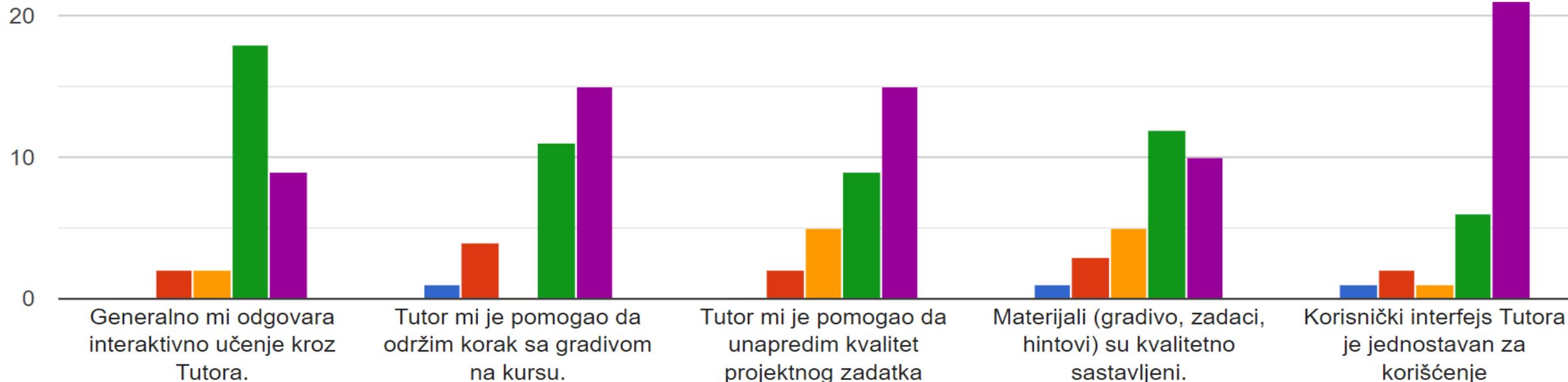


Ako naveden termin nije dostupan, sistem preporučuje dostupne termine u opsegu od nedelju dana od dana prvobitnog termina.

Pri uspešnom kreiranju termina, predavač treba da dobije mail potvrdu da će držati ispit.



1 - Veoma se ne slažem 2 - Ne slažem se 3 - Nemam mišljenje 4 - Slažem se 5 - Veoma se slažem



Godina	#	Learning experience	Redovnost rada	Doprinos projektu	Kvalitet materijala	Kvalitet interfejsa
2023	34	3.38	4.06	3.85	3.79	4.21
2024	31	4.1	4.13	4.19	3.87	4.42
Delta		0.72	0.07	0.34	0.08	0.21
Growth %		21.3%	1.72%	8.83%	2.11%	4.99%

Timestamp	[Generalno mi odgovara interaktivno učenje kroz Tutora.]	[Tutor mi je pomogao da održim korak sa gradivom na kursu.]	[Tutor mi je pomogao da unapredim kvalitet projektnog zadatka]	[Materijali (gradivo, zadaci, hintovi) su kvalitetno sastavljeni.]	[Korisnički interfejs Tutora je jednostavan za korišćenje]
16.4. 19:57:43	2 - Ne slažem se	1 - Veoma se ne slažem	2 - Ne slažem se	3 - Nemam mišljenje	1 - Veoma se ne slažem