

Kreacijski dizajn paterni

Singleton patern

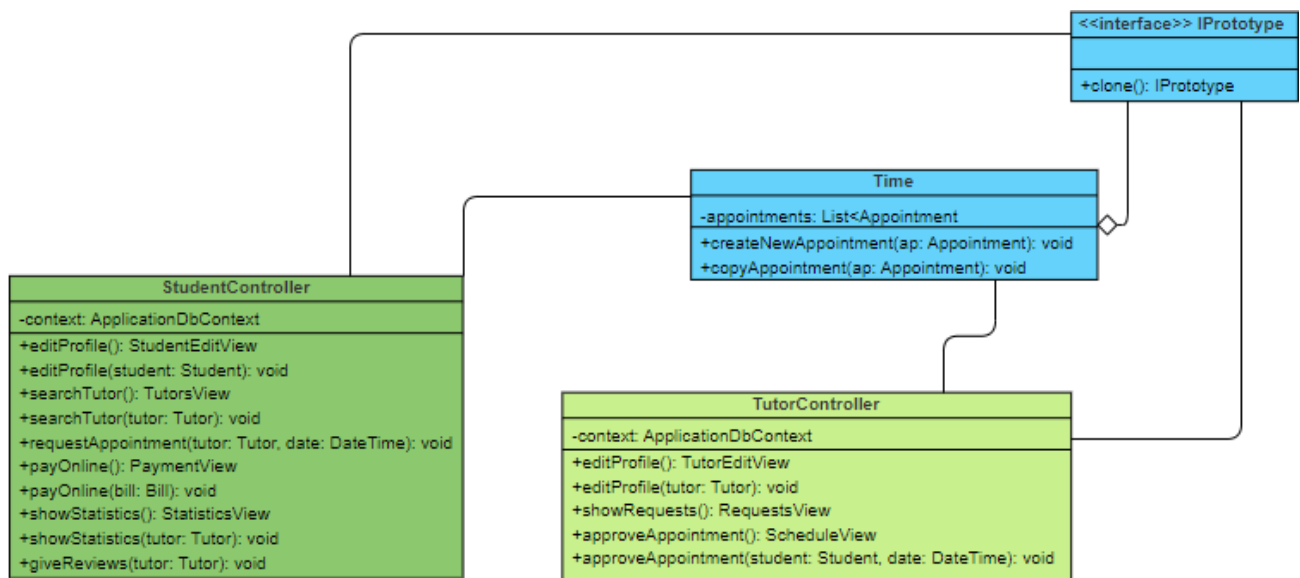
Singleton patern koristimo da bismo osigurali instanciranje klase samo jednom, a ujedno da imamo globalni pristup toj klasi.

Korisno bi nam bilo za upravljanje statistikama tutora, da nam klasa Statistics bude singleton i da ko god joj pristupa ima ažurne podatke i ne može ih mijenjati, nego se oni automatski izračunavaju svaki put kad se nešto doda ili obriše.

Prototype patern

Prototype patern dozvoljava da se kreiraju prilagođeni objekti bez poznavanja njihove klase ili detalja kako je objekat kreiran, tako što kloniramo i modifikujemo neki objekat, umjesto da kreiramo novi.

Možemo ga primijeniti na klasu Appointment, kada student želi da ugovori instrukcije sa istim tutorom na istom predmetu, samo drugi dan. To neće utjecati na prvobitno dogovorene instrukcije, nego će se one klonirati i modifikovati će se vrijeme održavanja.



Factory method patern

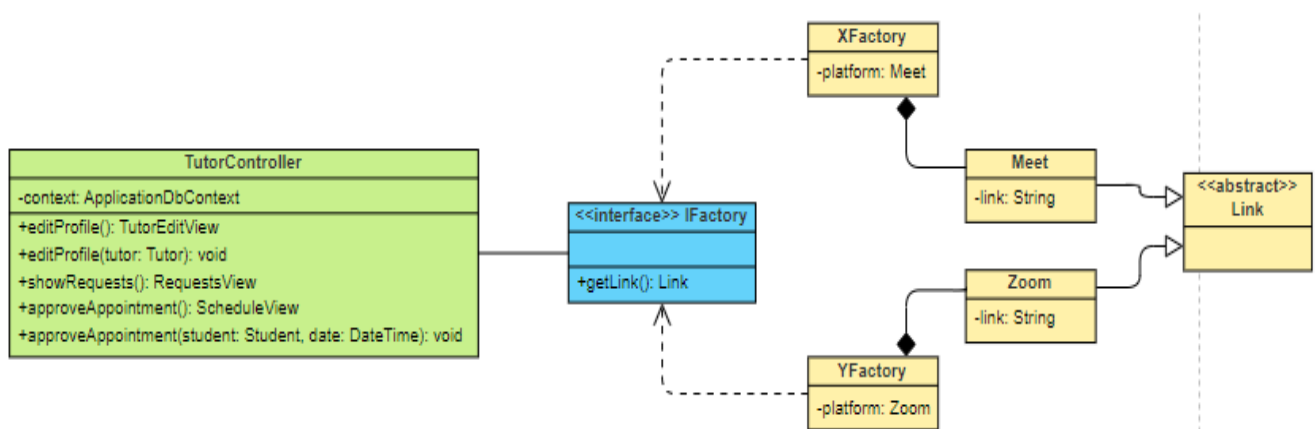
Factory method patern koristimo za kreiranje objekata na način da podklase odluče koju klasu instancirati, tako što instancira odgovarajuću podklasu preko posebne metode na osnovu informacije od strane klijenta ili na osnovu tekućeg stanja.

Primjer factory paternu možemo primijeniti u slučaju instruktora iz drugih zemalja, što je moguće zbog online načina održavanja instrukcija. Zavisno od zemlje u kojoj se instruktor nalazi, različite valute će zahtijevati, pa će tako i zavisiti od kursa prilikom pretvorbe iz naše valute u zahtijevanu.

Abstract factory patern

Abstract Factory patern omogućava da se kreiraju familije povezanih objekata bez specificiranja konkretnih klasa, tako što se definicija produkata odvoji od klijenta, tj. klijent ne zna stvarne tipove objekata kreiranih fabrikom, jer radi sa apstraktnim klasama i interfejsima.

U našem sistemu bi se mogao iskoristiti u slučaju online instrukcija da zavisno od toga šta instruktor koristi, student dobije gotov određen link npr. link za meet, od mogućih linkova meet, zoom, microsoft teams itd. koji su zapravo klase koje su izvedene iz abstraktne klase Link.



Builder patern

Builder patern koristimo za odvajanje specifikacije kompleksnih objekata od njihove stvarne konstrukcije, ako se objekti mogu podijeliti u skupove objekata koji se razlikuje samo po permutaciji njihovih dijelova.

Mogli bismo pojednostaviti klasu **Reviews**, tako da ima samo neophodne attribute: **Student** i **rate**, tj. da može da ostavi ocjenu, a da preko buildera omogućimo ostavljanje komentara ili nekih dodatnih informacija o sebi.