Išankstinė deklaracija - aprašyti, bet neapibrėžti. F-jų prototipai antraštės failuose

Abstrakčios klasės -Grynai virtualūs metodai neturi realizacijos. Klasė, turinti tokių metodų, yra abstrakti (negalima kurti jos objekto)

Agregacija - Vienos klasės objektas yra kitos klasės objekto dalis (bet ne atvirkščiai).

Asociacija - Vienos klasės objektas gali siųsti pranešimus kitos klasės objektams.

Enkapsuliacija - mechanizmas, susiejantis duomenis ir jiems taikomas operacijas, tokiu būdu (1) įgalinantis apsaugoti duomenis nuo nekorektiško panaudojimo (2) didinantis kodo atsparumą pokyčiams

Interfeiso sąvoka: Abstrakčiose klasėse gali būti dalinė (bendra) realizacija, nevirtualūs metodai, konstruktoriai, destruktoriai, etc. Interfeisu vadinama abstrakti klasė, kurioje nėra realizacijų.

Interfeisas - tai (viešų) metodų sąrašas, apibrėžiantis klasių bendravimo tarpusavyje būdus (kviestinų metodų prototipus).

Išankstinė deklaracija - aprašyti, bet neapibrėžti f-jų prototipai antraštės failuose...

**Konstruktoriai ir destruktoriai**

Daugialypio paveldėjimo atveju kompiliatorius užtikrina,   
kad išvestinio objekto viduje sukuriamas tik vienas virtualiai paveldėtos klasės egzempliorius,  
ir konstruktorius iškviečiamas tik vieną kartą.

Konstruktoriai:   
bazinių klasių konstr. kviečiami prieš išvestinės klasės konstr.  
anksčiausiai kviečiami virtualiai paveldėtų klasių konstruktoriai  
(tai atlieka „labiausiai išvestinė“ klasė)   
prieš kviečiant klasės konstruktorių kviečiami laukų konstruktoriai “lygiaverčiai“ konstruktoriai kviečiami ta tvarka, kokia yra nurodyta

Destruktoriai (virtualūs?):   
kviečiami atvirkščia tvarka nei konstruktoriai

Kompozicija - Vienos klasės objektas yra atsakingas už kitos klasės objekto gyvavimą.

Metodų perkrovimas - ta pati klasė, skirtingi prototipai.

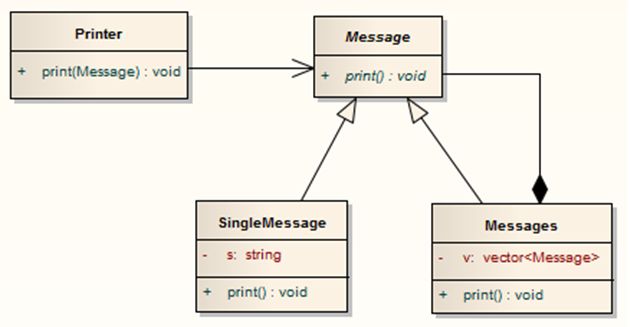
Metodų perrašymas - vienodi prototipai, skirtingos klasės (paveldi!)

Paveldėjimas - (klasių jungimo) būdas, kurio pagalba vienos klasės objektai gali įgauti kitos klasės objektų savybes.

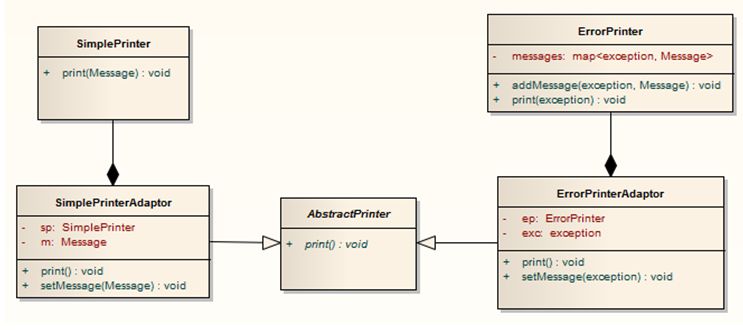
Polimorfizmas - galimybė tai pačiai operacijai veikti skirtingai skirtingose situacijose.

Virtualus copy/clone metodas Kopijos konstruktoriai enkapsuliuojami, o kopijavimas atliekamas polimorfizmo pagalba.

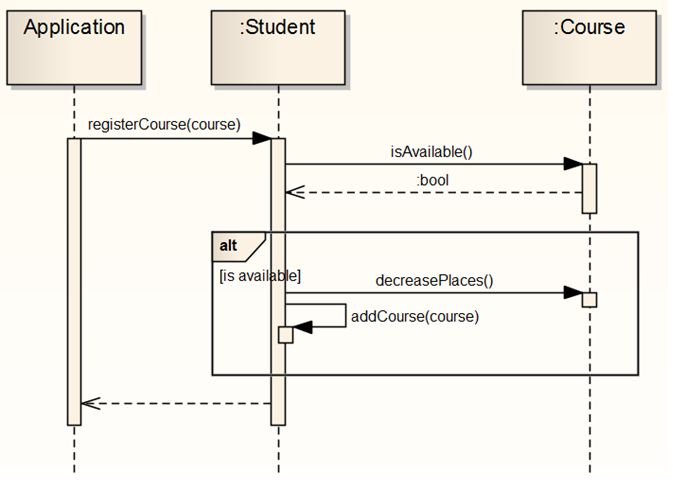
***Kompozicijai rekomenduoju Delegate, Adapter, Facade, paveldėjimui - Instance Method, kombinavimui - Proxy, Strategy, Composite, Decorator.***

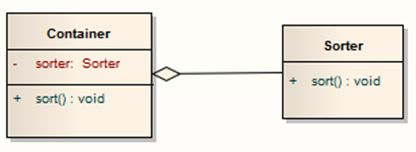


Composite šablonas - Naudojamas, kuomet norima objektų grupe operuoti taip pat, kaip pavieniu objektu. Apibrėžiamas pastovus interfeisas, kurį tenkina tiek komponuojama, tiek komponuojanti klasė

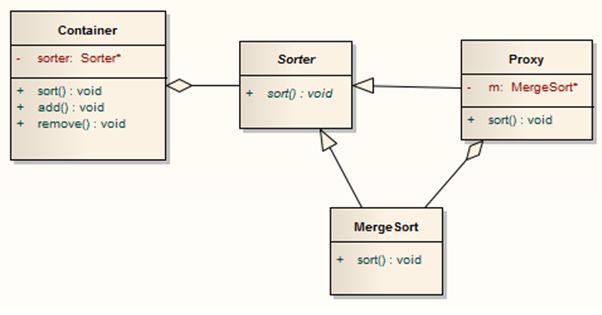


Adapter šablonas - Naudojamas, siekiant suderinti klases tarpusavyje. Apibrėžiamas „tarpinis“ objektas, atsakingas už „pritaikymą“.

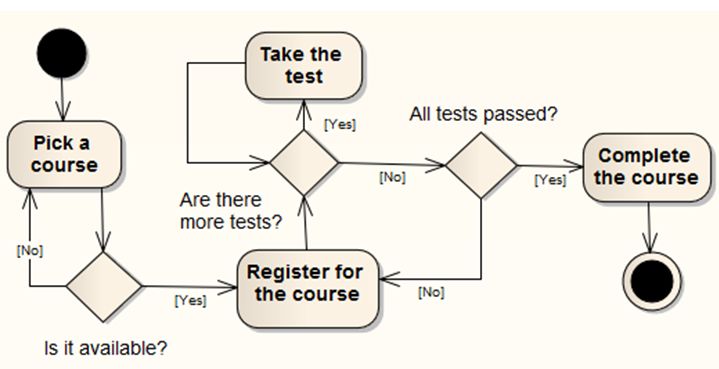
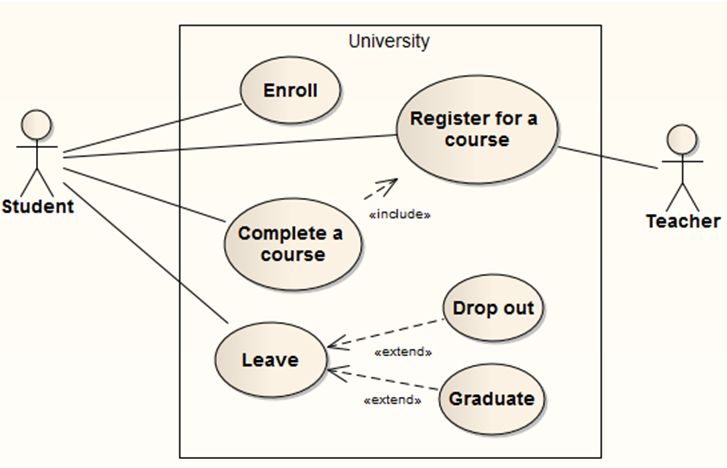




Delegate šablonas - Naudojamas siekiant paskirstyti atsakomybes. Objektas, vietoje to, kad atliktų veiksmą pats, deleguoja užduotį pagalbiniam objektui.



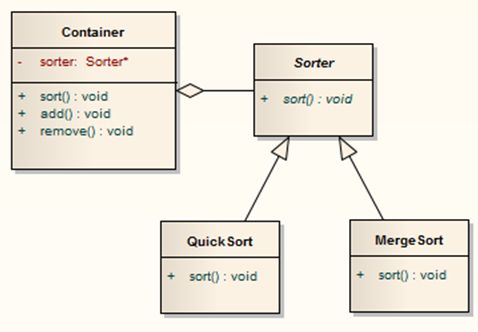
Proxy šablonas - Naudojamas, kuomet norima pateikti vieną objektą kito vietoje. Apibrėžiamas gaubiantis objektas, kuris deleguoja kreipinius komponuojamam objektui.



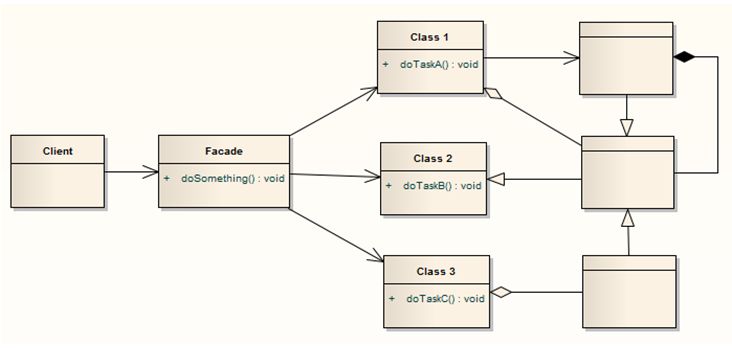
**Apie išimtis kuriant objektus** Išimtis konstruktoriuje - destruktorius tam objektui nebus iškviestas! Visi subobjektai, pilnai (!) sukurti iki išimties, naikinami.

**Throw sakinys** Korektiškas išimties konstruktoriuje naudojimas Throw be argumentų try bloke – nutraukia programą (iškviečia terminate). Throw be argumentų catch bloke – meta „einamąją“ išimtį (nekeičiant tipo)

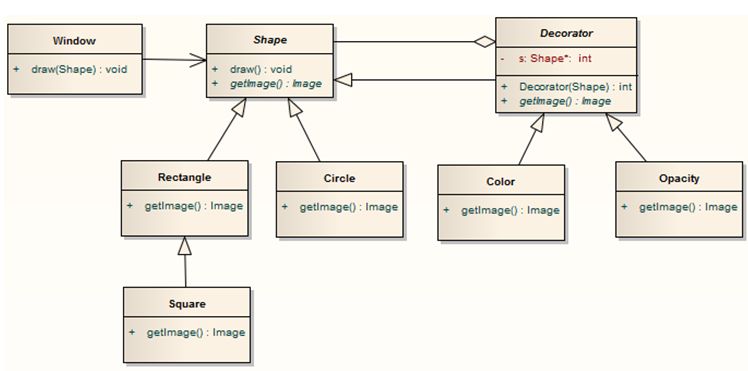
**Inicializatorių sąrašas** Galima gaudyti išimtis, kylančias kviečiant inicializatorius konstruktoriuje (angl. function try blocks). Apribojimai – kilus išimčiai subobjektų konstruktoriuose, konstruktorius negali paslėpti išimties (turi mesti). Galima pakeisti išimties tipą, taip pat atlikti „pašalinius“ veiksmus. Nenurodžius išimties, metama (rethrow) einamoji pagal nutylėjimą.



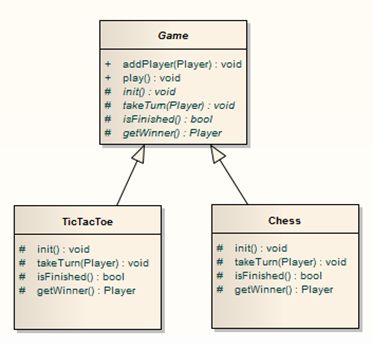
Strategy šablonas - Naudojamas, kuomet yra kintama taisyklė ar algoritmas. Apibrėžiamas pastovus interfeisas, įgalinantis vieną algoritmą keisti kitu.



Facade šablonas - Naudojamas, siekiant supaprastinti interfeisą. Sukuriama klasė, kurios realizacijoje paslepiami sudėtingi elementai.



Decorator šablonas - Naudojamas norint dinamiškai pridėti funkcionalumą. Apibrėžiama „dekoruojamą“ objektą gaubianti klasė, įgalinanti konstruoti „dekoracijų grandinę“.



Template method šablonas - Naudojamas kuomet turimas algoritmo skeletas, tačiau atskiri žingsniai gali keistis. Tų žingsnių apibrėžimas deleguojamas išvestinėms klasėms.