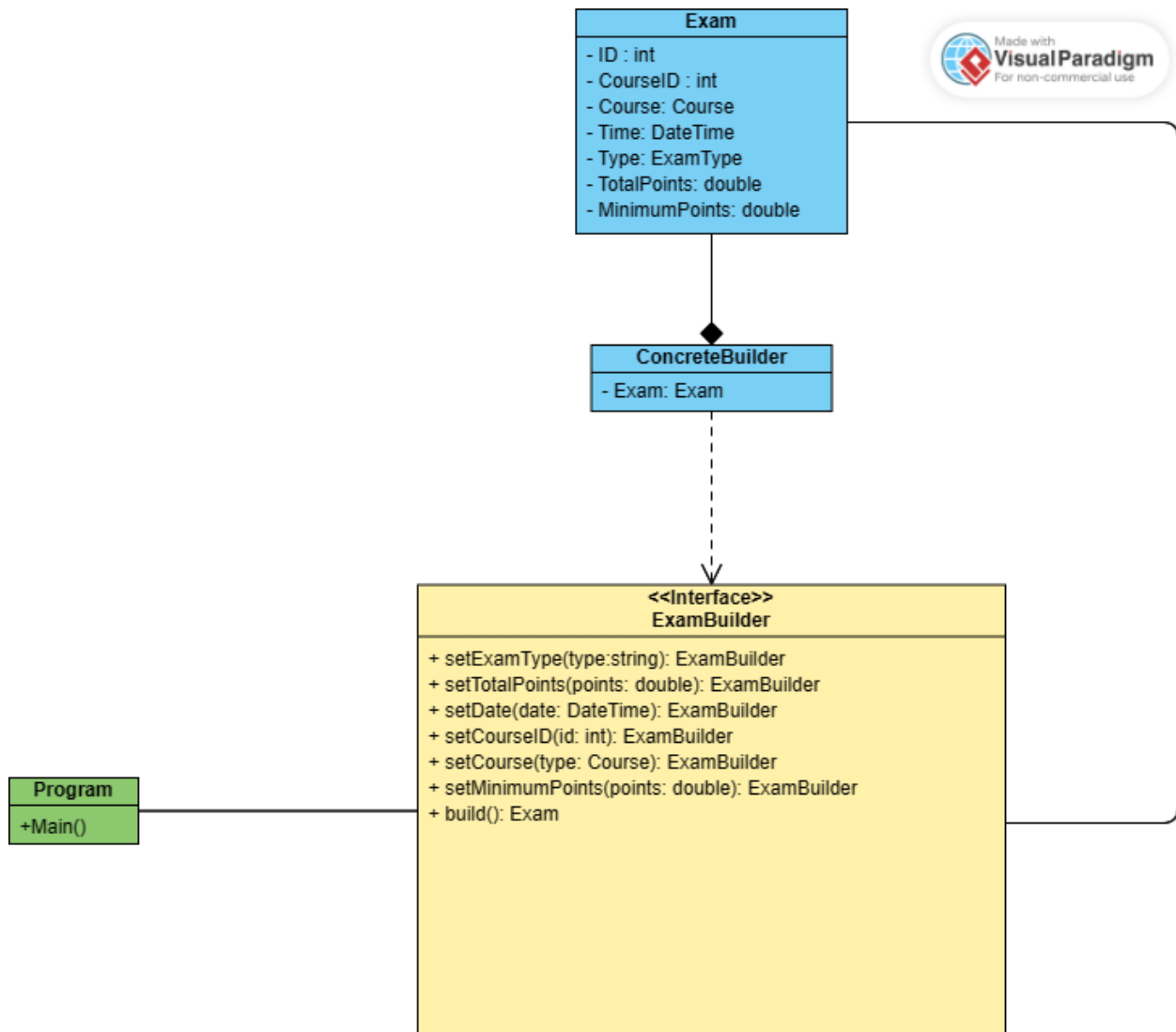


Kreacijski paterni

Builder

Ovaj patern može se koristiti ako imamo modele sa kompleksnom inicijalizacijom ili procesom konfiguracije, kao kod objekata Zadaća ili Ispit koji zahtijevaju više parametara. Ovo nam omogućava da kreiramo objekte korak po korak i dozvoljava nam da kreiramo drugačije konfiguracije za isti objekat.



Ovo također može biti primijenjeno na zadaću.

Prototip

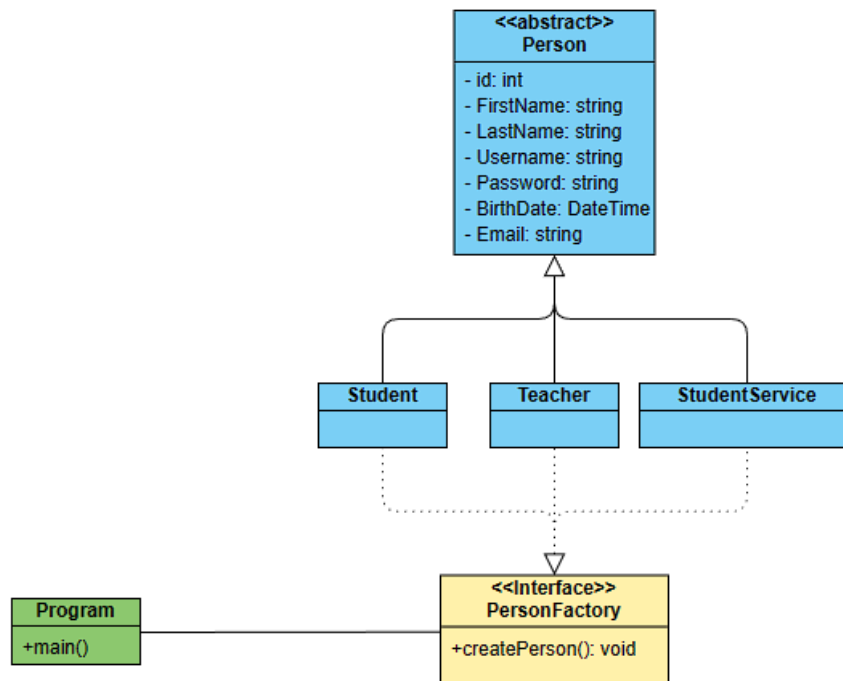
Koristi se za kreiranje novih objekata kloniranjem već postojećih objekata, čime izbjegavamo eksplicitno instanciranje. Korisno je kada imamo objekte koji su skupi za kreirati ili kada želimo kreirati nove objekte sa sličnim osobinama kao već postojeće.

Primjer primjene može biti kod Zadaća ili Ispita gdje ako imamo različite instance klasa koje dijele učestale atribute, kao naziv, datum, bodovi itd...

Factory

Factory pattern je korišten za kreiranje objekata bez razotkrivanja instancijske logike klientu. Ono daje interfejs ili baznu klasu za kreiranje objekata, i podklase ili implementacijske klase su odgovorne za kreiranje specifičnih instantaci.

Možemo primijeniti kod apstraktne klase Osoba gdje PersonFactory klasa enkapsulira kreacijsku logiku pomoću factory metode createPerson().



Abstract Factory

Abstract factory se koristi kada imamo skupinu povezanih ili zavisnih objekata, i želimo kreirati instance ovih objekata bez specificiranja njihove konkretne klase. Ovo pruža interfejs ili apstraktnu klasu za kreiranje porodice povezanih objekata, i svaka podklasa abstract factory je zadužena za kreiranje instanci specifičnih konkretnih klasa.

Singleton

Koristi se kada želimo osigurati da samo jedna instanca klase je kreirana kroz cijeli život programa, te želimo pružiti globalni pristup kroz tu instancu.

Primjer ovoga može biti uspostavljanje Singleton za pristup bazi, ako želimo da osiguramo da postoji samo jedna instanca odabranog kursa u sistemu, kreiramo *CourseManager* klasu kao Singleton.