

Dokumentacija - Zadaća 1: Ronilački klub

Nino Cerovec, 44444/15-R

Objašnjenje i razlozi implementacije uzoraka dizajna

Kako se radi o aplikaciji za ronjenje jednog kluba, implementirana je klasa Club, koja sadrži svojstva i metode vezane uz klub. Kako bi se osigurala jedinstvena reprezentacija kluba na razini cijele aplikacije, klasa Club implementira Singleton uzorak dizajna.

Implementirana je temeljna apstraktna klasa Agency koja implementira različite metode zajedničke za sve vrste agencija/federacija. Postoje različite agencije, no svi ronionci koji pripadaju istoj agenciji, dijele zajedničku agenciju u stvarnom svijetu i ne postoje različite reprezentacije iste agencije. Svaka vrsta agencije nasljeđuje klasu Agency i postavlja svoje vrijednosti za pojedine titule certifikata (klasa Level). Kako bi promjena na objektu agencije automatski utjecala na sve ronioce pripadnike iste agencije, želimo da dijele istu instancu objekta konkretne agencije. Zato svaka implementacija klase Agency još implementira i uzorak Singleton, jer nema smisla kreirati N jednakih objekata iste agencije za svakog pripadajućeg ronioca zasebno.

Implementirajući klasu Diver, uviđeno je da postoje različite vrste ronioca obzirom na agenciju/federaciju kojoj pripadaju, što je glavni parametar prema kojem se kreira objekt ronioca. Prema tome, implementiran je uzorak , kako bi bilo moguće na osnovu jednog parametra kreirati različite reprezentacije ronioca na osnovu temeljne apstraktne klase Diver i njezinih implementacija za različite agencije.

Imajući na umu programiranje usmjereno prema sučeljima, umjesto prema implementaciji, deklarirano je sučelje Algorithm, koje sadrži deklaraciju za jednu metodu makeDiveCombinations koja vraća kombinacije za uron na osnovu liste ronioca. Svaka vrsta algoritma za određivanje kombinacija ronioca implementira ovo sučelje.

Najzahtjevnija klasa bila je klasa urona (Dive) koja kombinira sve resurse ove aplikacije. Zbog složenosti klase i različitih čimbenika koji utječu na strukturu i prezentaciju pojedinog urona, jasno je kako je potreban segmentan pristup kod konstrukcije objekta ove klase. Zato je implementiran uzorak dizajna Builder koji omogućava da se ronionci i ronilačke kombinacije postavljaju zasebno, nakon što su generirani, ovisno o algoritmu koji se primjenjuje za generiranje.

Mogućnost i potreba implementacije uzorka dizajna Adapter na klasama ronilačkih kombinacija (DiveCombinations: DivePairs, DiveTroikas) se razmatrala, no zaključeno je da nasljeđivanje rješava isti problem u ovom slučaju, jer ne postoje metode klase koje je potrebno posebno prilagođavati, pa nema smisla nepotrebno komplicirati strukturu projekta.

