
PROGRAMSKI JEZICI 1

Zadatak 2 – Preklapanje operatora

Postavka

Kreirati projekat pod nazivom *Zadatak2* unutar rješenja (i.e. *Visual Studio solution*) pod nazivom *PJ1Zadaci2020*, te kreirati projekat pod nazivom *Zadatak2.Demo* unutar istog rješenja. Projekat *Zadatak2.Demo* treba da referencira biblioteku iz projekta *Zadatak2* i demonstrira njen rad.

Zadatak

Unutar projekta *Zadatak2*, kao statičku biblioteku, implementirati klase za predstavljanje i rad sa n -dimenzionim vektorima korištenjem dinamičke alokacije memorije, prema sljedećim stavkama.

- Implementirati klasu **VectorN** koja predstavlja dinamički alociran niz podataka tipa **double** i predstavlja vektor u n -dimenzionom vektorskom prostoru.
 - ↪ Omogućiti nasumični pristup elementima niza korištenjem odgovarajućih metoda. Obezbijediti imutabilnost objekata (individualni elementi vektora ne smiju se mijenjati).
 - ↪ Onemogućiti pristup izvan opsega niza proizvoljnim mehanizmom. Koristiti odgovarajući operator za pristup elementima.
 - ↪ Obezbijediti osnovne matematičke operacije za rad sa vektorima, te obezbijediti učitavanje i ispis vektora na standardni izlaz.
 - ↪ Obezbijediti instanciranje vektora korištenjem inicijalizatorske liste ili implicitnog kastovanja iz realnog broja.
 - ↪ Obezbijediti statičku metodu za konstruisanje nula-vektora proizvoljne dimenzionalnosti.
 - ↪ Obezbijediti matematički prikladne operatore poređenja.
 - ↪ Preklopiti operator funkcijskog poziva za izračunavanje vrijednosti polinoma predstavljenog vektorom.
- Implementirati klasu **ComparableVectorN** koja predstavlja n -dimenzioni vektor uporediv po specifičnom kriterijumu.
 - ↪ Obezbijediti konstruktor koji prihvata funkciju koja preslikava **VectorN** u **double** i **VectorN**, te implementirati odgovarajuće operatore poređenja upotrebom ove funkcije. Proslijeđena funkcija predstavlja funkciju za normiranje, pri čemu nije neophodno provjeravati njena svojstva pri konstrukciji.
 - ↪ Implementirati kastovanje iz klase **VectorN** u klasu **ComparableVectorN** uz korištenje podrazumijevane funkcije za normiranje koja predstavlja Euklidsko rastojanje od koordinatnog početka.
 - ↪ Implementirati kastovanje iz klase **ComparableVectorN** u klasu **VectorN**, uz zanemarivanje funkcije za normiranje.
 - ↪ Preklopiti operator dvosmjernog poređenja.
- Implementirati klasu **VectorSetN** koja predstavlja skup vektora iste dimenzionalnosti.
 - ↪ Obezbijediti umetanje i izbacivanje iz skupa korištenjem operatora dodjele sa dodavanjem i operatora dodjele sa oduzimanjem. Obezbijediti lančano pozivanje operatora.
 - ↪ Obezbijediti funkcije za transformaciju i filtriranje svih vektora iz skupa po nekom pravilu, korištenjem funkcije koja prihvata lambda operator.

- ~ Implementirati odgovarajuće funkcije za poređenje skupova (skupovi su identični ukoliko sadrže identične elemente).
 - ~ Implementirati ispis na standardni izlaz.
 - ~ Implementirati proveru linearne nezavisnosti vektora iz skupa.
- Implementirati klasu **OrderedVectorSetN** koja predstavlja skup uporedivih vektora, **ComparableVectorN**.
- ~ Obezbijediti umetanje i izbacivanje iz skupa korištenjem operatora dodjele sa dodavanjem i operatora dodjele sa oduzimanjem. Obezbijediti lančano pozivanje operatora.
 - ~ Implementirati odgovarajuće funkcije za poređenje skupova (skupovi su identični ukoliko sadrže identične elemente), te omogućiti poređenje objekata klasa **VectorSetN** i **OrderedVectorSetN**, tako da je operator poređenja komutativan.
 - ~ Omogućiti ispis na standardni izlaz u sortiranom poretku.
 - ~ Obezbijediti kastovanje iz skupa uporedivih vektora u skup vektora.
 - ~ Implementirati proveru linearne nezavisnosti vektora iz skupa.

Sve implementirane klase treba da poštuju pravilo trojke i pravilo petorke. Sve implementirane klase treba da budu objedinjene u jedinstven prostor imena, te treba da imaju ispravno razdvojena zaglavlja od implementacije. Pored toga, implementirane klase treba da obezbjeđuju mogućnost korištenja u obliku konstantnih objekata. Preklopiti operatore u svim situacijama kada je to semantički ispravno. Izbjeći svako dupliranje koda.

U projektu *Zadatak2.Demo* demonstrirati rad svih implementiranih klasa i metoda individualno.