

Programski jezici 1

Laboratorijska vježba br. 3 – Nasljeđivanje i polimorfizam

Priprema za laboratorijsku vježbu

Definisati interfejs sa podrazumijevanim ponašanjem kojim se obezbjeđuje ispis objekata na izlazni tok upotrebom operatora za ispis, bez potrebe da se operator preklapa za implementirajuće klase. Definisati interfejs kojim se omogućava poređenje objekata korištenjem relacionih operatora. Definisati interfejs kojim se predstavljaju 2D oblici koji se mogu porediti po obimu i čiji se relevantni podaci mogu ispisati na izlazni tok. Definisati interfejs za objekte koji se mogu pozicionirati u 2D prostoru i čije se koordinate mogu ispisati na izlazni tok, a koji se mogu porediti po njihovoj udaljenosti od koordinatnog početka.

Omogućiti rad sa pravougaonikima i kvadratima u 2D prostoru poravnatim sa koordinatnim osama. Omogućiti rad sa krugovima u 2D prostoru. Omogućiti rad sa proizvoljnim mnogouglovima (poligon sa proizvoljnim brojem tačaka) u 2D prostoru. Za sve ove oblike je moguće izračunati obim, moguće je da se porede po obimu, a moguće je i da se porede po udaljenosti od koordinatnog početka. Demonstrirati različite načine poređenja nekih od oblika. Oblici treba da implementiraju odgovarajuće interfejse.

Definisati klasu kojom se predstavlja neograničeni neuređeni skup elemenata, tako da element mogu biti proizvoljnog (iz perspektive korisnika klase) polimornog tipa iz ovog zadatka. Definisati klasu kojom se predstavlja neograničeni “sortirani” skup elemenata proizvoljnog polimornog tipa iz ovog zadatka, koji kao implementacioni detalj uvijek sadrži elemente u sortiranom poretku i omogućava brzu pretragu i brz ispis u sortiranom poretku. Definisati klasu kojom se predstavlja neograničeni “uporediv” skup elemenata proizvoljnog polimornog tipa iz ovog zadatka koji se mogu porediti po relaciji “je podskup” upotrebom relacionih operatora. Definisati klasu za predstavljanje skupa koji istovremeno ima karakteristike sortiranog skupa i uporedivog skupa. Za sve skupovne tipove važi da nemaju dupliciranih elemenata. Pravilno definisati metode i/ili operatore za: 1) dodavanje elementa u skup, 2) uklanjanje elementa iz skupa, 3) ispis svih elemenata skupa, 4) poređenje dva skupa po jednakosti, 5) filtriranje elemenata skupa, 6) izvršavanje akcije nad svim elementima skupa, 7) transformaciju elemenata skupa tako da slika svakog elementa može biti proizvoljnog polimornog tipa iz ovog zadatka (jedna metoda).

Napomene

- Pridržavati se principa objektno-orijentisanog programiranja i SOLID principa
- Pravilno omogućiti rad sa objektima u konstantnom kontekstu
- Za svaku operaciju omogućiti da se izvrši bez bočnih efekata i sa bočnim efektima (u mjestu)
- Pravilno izvesti enkapsulaciju, skrivanje informacija i modularizaciju
- Omogućiti duboko kopiranje i optimizacije performansi pri pomjeranju
- Pravilno omogućiti signalizaciju greške na mjestima gdje je to potrebno
- Pisati čitljiv kod, koristiti smislena imena i izbjeći dupliranje koda
- Pravilno izvršiti dinamičku alokaciju i dealokaciju memorije i izbjeći curenje memorije
- Korištenje STL struktura podataka i STL algoritama nije dozvoljeno