Programski jezici 1

Laboratorijska vježba br. 2 – Preklapanje operatora

Priprema za laboratorijsku vježbu

Definisati klasu **`Rectangle`** kojom se predstavlja pravougaonik poravnat s osama u 2D prostoru. Omogućiti da se pravougaonik može nalaziti bilo gdje u prostoru. Definisati sljedeće operatore:

- operator **+**, tako da zbir dva pravougaonika proizvodi njihov minimalni obuhvatajući pravougaonik, tj. pravougaonik minimalne površine unutar koga se nalaze oba pravougaonika (pri čemu je dozvoljeno da ivice novog pravougaonika dodiruju ivice pravougaonika operanada),
- komutativni operator ***** za skaliranje pravougaonika u odnosu na tačku presjeka dijagonala (centar pravougaonika), tako da se skaliranjem sa X dobije pravougaonik čija je površina X puta veća od površine pravougaonika operanda, s tim da odnos stranica pravougaonika treba da ostane isti i
- bitski AND operator (**&**), tako da vraća presjek pravougaonika operanada (takođe pravougaonik).

Prijedlog: Pri implementiranju operatora **+** i operatora **&**, razmotriti upotrebu maksimalnih i minimalnih vrijednosti koordinata tjemena po **x** i **y** osama.

Definisati klasu **`Set`** kojom se predstavlja neograničeni skup pravougaonika. Skup je neuređen i nema duplikata. Pravilno definisati operatore za: 1) dodavanje elementa u skup, 2) uklanjanje elementa iz skupa, 3) ispis svih elemenata skupa i 4) poređenje dva skupa po jednakosti. Pri ispisu, treba se ispisati niz uređenih trojki *(centar, stranica a, stranica b*). Omogućiti korištenje operatora u konstantnom kontekstu.

Rad u laboratoriji

Riješiti dobijeni zadatak, demonstrirati rad rješenja i priložiti rješenje.

Napomene

- Pridržavati se principa objektno-orijentisanog programiranja.
- Pravilno izvesti enkapsulaciju i skrivanje informacija.
- Pravilno izvršiti modularizaciju i razdvajanje interfejsa od implementacije.
- Omogućiti duboko kopiranje i optimizacije performansi pri pomjeranju.
- Pravilno omogućiti signalizaciju greške na mjestima gdje je to potrebno.
- Pisati čitljiv kod i koristiti smislena imena.
- Izbjeći dupliranje koda.
- Pravilno izvršiti dinamičku alokaciju i dealokaciju memorije.
- Izbjeći curenje memorije.
- Korištenje STL struktura podataka i STL algoritama nije dozvoljeno.