

Objektno programiranje

Prva zadaća – 4.4.2011

Napišite parametriziranu klasu **SET** koja predstavlja skup elemenata tipa T, te osnovne operacije nad skupovima.

Sami izaberite internu reprezentaciju skupa (lista, vektor itd.), ali obvezno iskoristite dinamičku alokaciju memorije (skup ne smije biti ograničene veličine).

Implementirajte:

1. konstruktore: prazan konstruktor, konstruktor koji stvara skup s jednim elemetom tipa T, konstruktor koji STL spremnik pretvara u skup elementa
2. destruktur
3. kontrolu kopiranja
4. operator pridruživanja
5. operator [] koji provjerava da li je objekt tipa T zadan u uglatim zagradama element skupa
6. operator + koji dodaje objekt tipa T u skup, odnosno koji vraća uniju skupova
7. operator - koji izbacuje objekt iz skupa, odnosno vraća razliku skupova
8. operator * koji vraća presjek skupova
9. funkciju size koja vraća broj elemenata u skupu
10. ispis skupova operatom << i unos skupova operatom >>. Naravno, za one tipove T za koje je definiran odgovarajući operator << tj. >>.
11. operator unarnog - koji vraća komplement skupa (univerzalni skup bez zadano skupa).
Budući da ne možete znati koji su sve objekti elementi univerzalnog skupa, razmislite kako biste zaobišli taj problem. Skup ne mora biti eksplicitno zadan nabranjem svojih elemenata.
Pripazite i na ispis ovakavih skupova.
12. operatore == i != koji ispituju jednakost odnosno različitost skupova
13. operatore <=, <, >= i > koji glume operacije podskupa i pravog podskupa
14. složene operatore pridruživanja

Napomene:

- Priilikom implementacije skupa **ne smijete koristiti STL spremnike**.
- Program se mora moći prevesti korištenjem Dev-C++ sučelja prema **gcc-u**.
- Zadaće morate poslati najkasnije do **četvrtka, 14.4.2011 do 23:59** sati na email adresu **nuno(at)math.hr**
 - Sva eventualna pitanja vezana uz prvu zadaću šaljite na istu email adresu
- Zadaćama će biti dodijeljeni bodovi koje zatim trebate **obraniti u subotu, 16.4.2011** odmah nakon prvog kolokvija
 - Obrana će trajati oko 30 minuta, a sastoji se od toga da ćete u praktikumu trebati napraviti malu izmjenu u vlastitom programskom kodu.
 - Ukoliko izmjenu znate napraviti, dokazali ste da ste sami pisali zadaću (ili da je barem u dovoljnoj mjeri razumijete)
 - Ukoliko ne uspijete napraviti izmjenu, dobit ćete 0 bodova iz zadaće, što ustvari povlači pad kolegija.
 - Dakle, bolje je da sami napišete nepotpuno rješenje zadaće i da zatim s lakoćom obranite bodove koje ste dobili za nepotpuno rješenje, nego da vam netko drugi napiše potpuno rješenje za koje ćete dobiti sve bodove koje nećete moći obraniti.
- Možda nećete biti uhvaćeni u varanju, ali ako vas uhvatimo, definitivno padate kolegij, uz prijavu prodekanu za nastavu. Istu posljedicu imaju i sva "**preslična**" rješenja.