
PROGRAMSKI JEZICI 1

Zadatak 1 – Enkapsulacija i skrivanje informacija

Postavka

Kreirati projekat pod nazivom *Zadatak1* unutar rješenja (i.e. *Visual Studio solution*) pod nazivom *PJ1Zadaci2020*, te kreirati projekat pod nazivom *Zadatak1.Demo* unutar istog rješenja. Projekat *Zadatak1.Demo* treba da referencira biblioteku iz projekta *Zadatak1* i demonstrira njen rad.

Zadatak

Unutar projekta *Zadatak1*, kao statičku biblioteku, implementirati klase za rad sa dinamičkom memorijom, prema sljedećim stavkama.

- Implementirati klasu **Array** koja predstavlja dinamički alociran niz podataka tipa **unsigned char**.
 - ↪ Omogućiti nasumični pristup elementima niza korištenjem odgovarajućih metoda. Omogućiti modifikaciju elemenata korištenjem referenci.
 - ↪ Onemogućiti pristup izvan opsega niza korištenjem signalizacije o grešci kroz neki parametar prosljeđen po referenci.
 - ↪ Obezbijediti osnovne metode za informisanje korisnika o veličini niza.
 - ↪ Veličina niza treba da bude specifikovana kroz konstruktor i potrebno je onemogućiti implicitno kastovanje iz cijelog broja u niz.
 - ↪ Obezbijediti metodu za poređenje dvaju nizova posmatrajući ih kao velike brojeve u bazi 256.
- Implementirati klasu **Stack** koja predstavlja stek podataka proizvoljnog tipa.
 - ↪ Klasa treba interno da koristi ručno implementiranu uvezanu listu definisanu kao slog enkapsuliran u klasi **Stack**.
 - ↪ Za predstavljanje proizvoljnog tipa koristiti univerzalni pokazivač, **void***.
 - ↪ Pri kreiranju kopija dozvoljeno je kopiranje istog pokazivača. Destruktori ne treba da uništavaju originalni podatak. Klasa **Stack** koristi pokazivače na eksterni objekat umjesto njihovog kopiranja.
 - ↪ Obezbijediti metode **void push(void*)**, **void* pop()** i **void* top()** za umetanje, izbacivanje i prikazivanje vrha, respektivno. Vratiti **nullptr** ukoliko je stek prazan. Onemogućiti dodavanje **nullptr** vrijednosti.
- Implementirati klasu **BinaryTree** koja predstavlja binarno stablo podataka tipa **Array**.
 - ↪ Obezbijediti metode za umetanje u stablo (metode za brisanje čvorova nisu neophodne). Nakon umetanja, stablo treba da obezbjeđuje sortiran poredak. Pri tome, poređenje vrijednosti se vrši korištenjem pokazivača na funkciju koji je nestatički član klase **BinaryTree** i koji se ne može mijenjati nakon što je **BinaryTree** kreiran.
 - ↪ Klasu za čvorove enkapsulirati u klasu **BinaryTree**.
 - ↪ Napisati statičku metodu u klasi **BinaryTree** koja obilazi stablo u inorder poretku, korištenjem implementirane klase **Stack** (bez korištenja rekurzije u metodi). Ova metoda treba da prihvati, kao argument, pokazivač na funkciju koja će biti pozivana pri inorder obilasku uz prosljeđivanje podatka sa trenutnog čvora (e.g. funkcija se može koristiti za ispis čvorova). Implementirati i istoimenu metoda bez argumenata, koja ispisuje stablo u inorder poretku na standardni izlaz, tako da vrijednost svakog čvora bude predstavljena kao veliki broj u heksadecimalnom zapisu.

Sve implementirane klase treba da sadrže odgovarajuće konstruktore i destruktore, te omogućavaju optimizacije performansi pri pomjeranju objekata. Sve implementirane klase treba da budu objedinjene u jedinstven prostor imena, te treba da koriste ispravno razdvojena zaglavlja od implementacije. Pored toga, implementirane klase treba da obezbjeđuju mogućnost korištenja u obliku konstantnih objekata. Ukoliko je to neophodno, izvesti različite implementacije istih metoda za korištenje u LVALUE i RVALUE kontekstima.

Implementaciju je neophodno uraditi uz izričito pridržavanje principa objektno-orijentisanog programiranja, pridržavanje opštih principa programiranja i pisanja čistog koda, kao i pridržavanje konvencija za programski jezik C++. Ispravno koristiti enkapsulaciju i skrivanje informacija, te izbjeći dupliranje koda. Dozvoljeno je, i preporučuje se, pisanje pomoćnih klasa i metoda, pored onih koji su traženi u zadatku, kad god ono služi čitljivosti, održivosti ili efikasnosti koda.

U projektu *Zadatak1.Demo* demonstrirati rad svih implementiranih klasa i metoda individualno.