
PROGRAMSKI JEZICI 1

Zadatak 5 – Generički mehanizam

Postavka

Kreirati projekat pod nazivom *Zadatak5* unutar rješenja (i.e. *Visual Studio solution*) pod nazivom *PJ1Zadaci2020*, te kreirati projekat pod nazivom *Zadatak5.Demo* unutar istog rješenja. Projekat *Zadatak5.Demo* treba da referencira biblioteku iz projekta *Zadatak5* i demonstrira njen rad.

Zadatak

Unutar projekta *Zadatak5*, kao statičku biblioteku, implementirati baferisanu kolekciju prema sljedećim uputstvima, uz korištenje dinamičke alokacije memorije.

- Definirati metaklasu koja predstavlja interfejs, a potom definisati interfejs **IPrintable** koji omogućava ispis objekata klase **IPrintable** na standardni izlaz korištenjem odgovarajućeg operatora i virtuelne metode.
- Implementirati klasu **BufferedCollection<T>** koja predstavlja neuređenu kolekciju elemenata tipa **T**, tako da je nad objektima tipa **T** moguće vršiti operacije poređenja, sabiranja i ispisa na standardni izlazni tok, kao i kopiranja i premiještanja. Onemogućiti prevođenje kada ovi uslovi nisu zadovoljeni. Koristiti koncepte, ograničenja i statičke pretpostavke pri realizaciji rješenja.
- Klasa **BufferedCollection<T>** treba da obezbijedi odgovarajuće funkcije za umetanje, izbacivanje i pristup elementima kolekcije.
- Omogućiti ispis svih elemenata kolekcije u sortiranom poretku na standardni izlaz uz implementaciju interfejsa **IPrintable**.
- Klasa treba da koristi uvezanu listu pokazivača na dinamički alocirane nizove veličine proslijeđene kroz konstruktor.
 - ↪ Nizovi na koje pokazuju elementi uvezane liste predstavljaju baferu u kojima se čuvaju podaci. Ukoliko je u baferisanoj kolekciji umetnuto n elemenata, a veličina bafera je k , pri čemu je $k < n$ tada će trenutni broj elemenata liste da bude $\left\lceil \frac{n}{k} \right\rceil$, gdje će svaki da pokazuje na niz dužine k . Pristup elementu sa indeksom iz opsega $[0, k - 1]$ izvešće se nad prvim baferom u listi, dok će pristup elementu sa indeksom iz opsega $[k, 2k - 1]$ biti izveden nad drugim baferom u listi. Na ovaj način, baferisana kolekcija je segmentisana u podnizove dužine k .
 - ↪ Pri dodavanju novog elementa u kolekciju, ukoliko je neophodno proširivanje bafera, izvršiti sve potrebne alokacije memorije uz obradu i dalju propagaciju izuzetaka pri pristupu memoriji.
 - ↪ Pri izbacivanju elemenata, nije neophodno smanjivati bafer, ali je neophodno pomjeriti preostale elemente.
- Omogućiti određivanje sume vrijednosti u kolekciji.

Sve implementirane klase treba da poštuju pravilo trojke i pravilo petorke, budu objedinjene u jedinstven prostor imena i imaju ispravno razdvojena zaglavlja od implementacije. Preklopiti operatore u svim slučajevima kada je to semantički ispravno. Izbjeći svako dupliranje koda. Omogućiti podizanje i obradu izuzetaka u slučajevima kada je to neophodno, te ispravno označiti sve metode s obzirom na korištenje izuzetaka i konstantni kontekst.

U projektu *Zadatak5.Demo* demonstrirati rad svih implementiranih klasa i metoda individualno. Ilustrovati sve slučajeve kada dolazi do bacanja izuzetka i obraditi ove izuzetke. Pri izvršavanju programa ne smije doći do poziva funkcije **terminate()**.