
PROGRAMSKI JEZICI 1 (3.7.2017.)

1. Napisati klasu **FragmentList<T>** koja predstavlja dinamičku listu elemenata tipa **T**. Ova lista treba da zadovolji sljedeće uslove.
 - Klasa treba da koristi dinamičku alokaciju, bez korištenja standardne biblioteke, pri čemu korisnik klase vrši alokaciju manuelno, pozivom odgovarajuće metode, tako da su čuvani elementi raspoređeni u listi nizova (pogledati naredne stavke).
 - Kompajliranje je dozvoljeno samo u slučajevima kada je tip **T** aritmetički.
 - Klasa treba da se ponaša polimorfno (pogledati 2. zadatak).
 - Lista treba da interno koristi kolekciju pokazivača na nizove tipa **T**, a vrši alokaciju tih nizova kroz metodu **allocate(int n)** koju korisnik klase poziva.
 - Lista treba da ima metodu za dodavanje novog elementa, a ukoliko u listi nema mjesta za nove elemente, treba da podigne osnovni izuzetak iz standardne biblioteke sa odgovarajućom porukom.
 - Lista treba da ima metodu koja vraća uređeni par sa trenutnim ukupnim brojem zauzetih elemenata i trenutnim brojem slobodnih elemenata.
 - Lista treba da implementira operator indeksiranja koji pristupa elementima redom njihovog dodavanja u listu.
 - Optimizovati klasu tako da se konstruktor kopije ne aktivira nad privremenim objektima.
 - Pri implementaciji klase izbjeći direktno korištenje petlji.
2. Napisati klasu **AutomaticFragmentList<T>** izvedenu iz **FragmentList<T>**. Ova klasa treba da redefiniše metodu za dodavanje, tako da se pokušava dodavanje pozivom metode iz natklase, a ukoliko ona baci izuzetak, alocira se novi fragment pozivom metode **allocate(int n)** iz natklase, bez bacanja izuzetka.
 - 2.1. Ukoliko je u ovoj metodi uhvaćen izuzetak koji baca metoda natklase, može li se desiti da ova metoda ipak baci izuzetak. Ukoliko je odgovor da, koji je to izuzetak?
3. Napisati primjer C++ programa (jedan izvorni fajl, „*main.cpp*“) koji učitava karaktere iz fajla pod nazivom „*input.txt*“ i upisuje ih u obrnutom redoslijedu u fajl pod nazivom „*output.txt*“. Dozvoljeno je, i preferira se, korištenje standardne biblioteke.
4. Odgovoriti na sljedeće stavke.
 - 4.1. Napisati primjer zapisa tipa RVALUE reference.
 - 4.2. Napisati primjer polimorfnog ponašanja korištenjem reference.
 - 4.3. Napisati primjer korištenja foreach petlje za iteriranje kroz instancu klase `std::vector<double>`.
 - 4.4. Na dva načina definisati generičku metodu **add(const A&, const B&)** za množenje dva podatka klase **A** i **B**, podrazumijevajući da postoji operator množenja nad njima.
 - 4.5. Napisati funkciju **pw(int k, int n)** koja izračunava vrijednost k^n , tako da se izračunata vrijednost dobija za vrijeme kompajliranja, te da se njena povratna vrijednost može iskoristiti kao argument šablona (koristi se C++11 standard).

NAPOMENE:

- Minimizirati broj linija koda i izbjeći dupliranje koda.
- Ispravno označiti metode koje sigurno bacaju izuzetke i koje sigurno ne bacaju izuzetke, kao i konstante metode.
- Poštovati konvencije, razdvojiti interfejs od implementacije.