

Oduzimanje polinoma.

Implementirati metodu `Polynomial Polynomial::sub(Polynomial&);`.
Metoda oduzima dva polinoma i vraća novi objekat klase.

Množenje polinoma.

Implementirati metodu `Polynomial Polynomial::mul(Polynomial&);`.
Metoda množi dva polinoma i vraća novi objekat klase.

Deljenje polinoma.

Implementirati metodu `Polynomial Polynomial::div(Polynomial&);`.
Metoda deli dva polinoma i vraća novi objekat klase koji predstavlja rezultat deljenja dva polinoma odbacujući ostatak pri deljenju.

Vrednost polinoma u tački.

Implementirati metodu `double Polynomial::value(double& x);`.
Metoda vraća vrednost polinoma u tački x .

Poređenje polinoma (equals).

Preklopiti operator `==` za upoređivanje dva polinoma, gde je prototip operacije za upoređivanje `bool operator==(const Polynomial&, const Polynomial&)`. Operatorska funkcija vraća vrednost `true` ukoliko su polinomi p_1 i p_2 jednaki.

Ispisivanje polinoma u izlazni tok.

Omogućiti ispis polinoma u izlazni tok. Implementirati metodu `print` koja ispisuje polinom po formatu koji odgovara formatu ulazne niske prilikom kreiranja polinoma. Uz član sa eksponentom 1 ne ispisivati komponentu `"^1"`. Uz član sa eksponentom 0 ne ispisivati komponentu `"x^0"`. Uz član kojeg množi koeficijent 1, ne ispisivati koeficijent, osim ukoliko je u pitanju slobodan član polinoma.

```
void print(ostream& out, Polynomial& poly);
```

Test funkcija

U projektu glavnog programa treba da postoji funkcija `void test();` koja testira rad sa polinomima. Studenti treba da implementiraju datu funkciju i uslovno je prevode ako je definisan makro `STUDENT_TEST`. Predvideti i postojanje makroa `PROF_TEST`. Ako su oba makroa definisana, `PROF_TEST` ima prioritet i tada se prevodi tajni test primer koji će biti dat na odbrani. Funkcija `main` treba samo da pozove test funkciju.

Tehnički zahtevi i smernice za izradu rešenja

Sve klase i metode moraju biti imenovane prema zahtevima iz domaćeg zadatka. Svaka klasa koja koristi dinamičku memoriju mora biti bezbedna za korišćenje. Programski kod klasa rasporediti u odgovarajuće `.h` i `.cpp` fajlove. Nije dozvoljeno korišćenje globalnih promenljivih za razmenu podataka. Ukoliko u zadatku nešto nije dovoljno jasno definisano, treba usvojiti razumno pretpostavku i na temeljima te pretpostavke nastaviti izgrađivanje svog rešenja.

VAŽNE NAPOMENE

Za uspešno odbranjen domaći zadatak potrebno je na odbrani pokazati kod podeljen na odgovarajuće projekte, `.h` i `.cpp` fajlove.

- Klase kojima su implementirani osnovni koncepti treba da budu smeštene u poseban projekat rešenja koji se prevodi kao statička biblioteka (`Polynomial.lib`).
- Glavni program napisati u posebnom projektu koji se prevodi kao Win32 Console Application (`testdz1.exe`) fajl i koji treba povezati sa statičkim bibliotekama. Glavni program treba implementirati tako da pozove globalnu funkciju `void test();`
- Studenti treba da implementiraju svoju verziju ove funkcije tako da demonstriraju operacije sa polinomima.
- NIJE DOZVOLJENO SMESTITI CEO KOD U JEDAN PROJEKAT ILI CPP fajl!