

Naziv projekta: POLINOM

Predmet: Strukture podataka i algoritmi

Predmetni asistent: Beširević Admir

Student: Čajić Dino

Broj indeksa: 122-IT/19

Sadržaj:

1. Uvod.....	3
2. Konstruktori.....	4
3. Funkcije.....	5
4. Preklopljeni operatori	6

1. Uvod

Klasa Polinom je implementirana kao dvosmjerna lista. Sastoji se od čvorova koji zapravo predstavljaju monome.

Jedna od prednosti ovog načina implementacije u poređenju sa naprimjer implementacijom preko vektora jeste ta što možemo izbjeći čuvanje koeficijenata koji su jednaki nuli. Primjer slučaja gdje bi tako nešto morali uraditi jeste unos polinoma $x^{999}+1$.

Svaki monom ima koeficijent koji je tipa double, stepen koji je tipa int i dva pokazivača (pokazivač na sljedeći čvor i pokazivač na prethodni čvor).

Lista odnosno Polinom ima dva obavezna pokazivača, tj. glava koji je pokazivač prije početka liste i rep koji je pokazivač iza kraja liste. Njihova uloga nam daje garanciju da nikada nećemo se naći izvan opsega, te nam olakšava implementaciju nekih funkcija. Lista također sadrži unutar sebe trenutni broj monoma/elemenata.

2. Konstruktori

Void init() - Void funkcija koja se koristi u svakom konstruktoru za „osnovnu inicijalizaciju“, odnosno definiše glavu, rep, prilikom čega postavlja da rep slijedi glavi, a glava prethodi repu, i postavlja broj elemenata (monoma) na 0.

Polinom() - Konstruktor bez parametara koji samo poziva funkciju init().

Polinom(double n, int m) - Konstruktor sa dva parametra (n tipa double koji predstavlja koeficijent i m tipa int koji predstavlja stepen). Konstruktor poziva funkciju init(), te kreira jedan monom sa koeficijentom n i stepenom m.

Polinom(const vector<double> &k, const vector<int> &s) - Konstruktor sa dva parametra (vektor k koji u sebi čuva elemente tipa double i vektor s koji u sebi čuva elemente tipa int). Konstruktor poziva funkciju init(), te kreira polinom koji se sastoji od broja čvorova (monoma) koliko vektor k ima elemenata. Koeficijenti monoma će postati vrijednosti vektora k, dok će stepeni monoma postati vrijednosti vektora s.

Polinom(const Polinom &p) - Konstruktor kopije, vrši duboko kopiranje polinoma p u novo-kreirani polinom.

~Polinom() - Destruktor, briše sve monome, te postavlja glavu i rep na nullptr.

3. Funkcije

void Dodaj(double k, int s) - Void funkcija koja prima dva parametra (k tipa double koji predstavlja koeficijent i s tipa int koji predstavlja stepen). Funkcija dodaje na polinom na koji je pozvana novi monom sa koeficijentom k i stepenom s. Dodaje novi čvor na kraj polinoma tako što pokazivač na sljedeći od zadnjeg elementa preusmjeri da pokazuje na novokreirani čvor kojem prethodi upravo taj, a slijedi rep polinoma, te samo još preusmjeri pokazivač na prethodni čvor od repa na taj novokreirani čvor.

void Ukloni() - Void funkcija koja uklanja monom sa početka polinoma. To radi tako što pokazivač na sljedbenika glave preusmjeri da pokazuje na drugi čvor u polinomu, dok drugom čvoru pokazivač na prethodni čvor preusmjeri da pokazuje na glavu i samo obriše monom koji je bio između njih.

void sortiraj() - Void funkcija koja sortira polinom na koji je pozvana u odnosu na stepen monoma u rastućem poretku. To znači da će polinom (ukoliko ima slobodni član) nakon sortiranja imati slobodni član na prvom mjestu jer slobodni član ima stepen 0 koji naravno ne pišemo. Ova funkcija radi tako što napravimo dva vektora u koje smještamo koeficijente i stepene polinoma. Sortiramo ih po stepenu tj. mijenjamo pozicije stepena ali i koeficijenata u vektorima, zatim kreiramo novi polinom pomoću konstruktora kojem proslijedimo oba vektora i postavimo da je „trenutni“ polinom taj novo-kreirani.

void preuredi() - Void funkcija koja sabere/oduzme monome unutar polinoma tako da dobijemo „lijep“ oblik polinoma tj. da se određeni stepen pojavljuje samo jednom. Ona funkcionira tako što prvo pozove funkciju

sortiraj(), zatim prolazi kroz polinom for petljom i uspoređuje da li su istog stepena trenutni monom i prethodni. Ako jesu, koeficijent prethodnog se dodaje na trenutni dok se prethodni briše.

int velicina() - Trivijalna funkcija koja vraća broj elemenata(monoma).

bool prazno() - Trivijalna funkcija koja vraća „true“ ako je broj elemenata jednak nuli, a „false“ u suprotnom.

Monom* pocetak() - Trivijalna funkcija koja vraća pokazivač na monom koji se nalazi na prvom mjestu, odnosno odma iza glave.

4. Preklopljeni operatori

Polinom& operator =(Polinom &p) - Operator dodjele.

Polinom& operator += (Polinom &p) - Operator += funkcioniše tako što se dodaju pomoću funkcije Dodaj(double m, int n) svi monomi polinoma p na polinom na koji se pozvao operator te se pozove funkcija preuredi().

Polinom& operator -= (Polinom &p) - Operator -= je implementiran isto kao i operator += samo što pri dodavanju monoma dodaje se koeficijent sa suprotnim predznakom.

Npr. $(2+3x) - (2x + 2) = 2+3x + (-2x)-2$

Polinom& operator *= (Polinom &p) - Operator *= funkcioniše tako što prolazimo sa for petljom sve monome „trenutnog“ polinoma tj. polinom na kojem se treba desiti promjena i unutar te for petlje prolazimo drugom for

petljom sve monome drugog polinoma. Svaki se monom jednog polinoma pomnoži sa svakim monomom drugog polinoma i to dodajemo u novi Polinom. Na kraju samo stavimo da je „trenutni“ polinom jednak novom, te pozovemo `preuredi()`.

Polinom& operator ~ () - Operator `~` predstavlja izvod. On funkcionise tako što se svakom monomu koeficijent pomnoži sa stepenom te to postavi kao novi koeficijent i stepen umanji za jedan.

Polinom& operator & () - Operator `&` predstavlja integral. On funkcionise tako što svakom monomu poveća stepen za jedan te koeficijent podijeli sa stepenom i to postavi kao novi koeficijent.

friend ostream& operator << (ostream& cout, Polinom &p) - Prijateljska funkcija za operator `<<`. Ona ispisuje polinom u ispravnom i „najljepšem“ obliku gledajući različite uslove.

friend istream& operator >> (istream& cin, Polinom &p) - Prijateljska funkcija za operator `>>`. Ona omogućava unos polinoma preko tastature. Prvo ukloni sve monome iz polinoma nad kojim je pozvan ovaj operator, zatim se koristi evaluator da bi provjerio je li validan izraz koji je korisnik unio, ali i da bi izračunao tu vrijednost u isto vrijeme. Ukoliko je sve prošlo bez problema izračunati polinom se smješta u polinom nad kojim je pozvan operator.