# Projekat 1: Polinom

IDEJA:

Klasa polinom sadrži svoju strukturu Clan i potklasu Iterator. Unutar strukture Clan čuvamo koeficijent i stepen svakog člana (npr. za član 2x^5 koeficijent=2, stepen=5), te pokazivače na prethodni i sljedeći član. Imamo i konstruktor sa 4 parametra (stepen, koeficijent, sljedeci, prethodni) koji postavlja sve vrijednosti datog člana. Osim toga, implementirane su i dvije funkcije VratiStepen i VratiKoeficijent za koje je očigledno šta rade. Što se tiče potklase Iterator, ona funkcioniše kao standardni iterator. Podržane su unarne operacije preincrement (vraća iterator na sljedeći član polinoma), postincrement (vraća trenutni iterator te ga nakon toga povećava za jedan-prebacuje na sljedeći član), predecrement (vraća iteraror na prethodni član polinoma) i postdecrement(vraća iterator na trenutni član te ga prebacuje na pretodni), kao i operator dereferenciranja. Isto kao i struktura Clan, i potklasa Iterator posjeduje funkcije VratiStepen i VratiKoeficijent kao i operatore == i !=. Sama klasa Polinom je implementirana kao dvostruko povezana lista, koja ima pokazivače na član head (član prije prvog člana), na član tail (član poslije posljednjeg člana), te jednu varijablu tipa int koja čuva broj članova datog objekta tipa Polinom. Zamišljeno je da ova klasa podržava umetanje i brisanje članova, prolazak iteratorom kroz listu, te preklopljene operatore +, -, \*, ^ koji služe za sabiranje, oduzimanje, množenje i stepenovanje objekata tipa Polinom.

FUNKCIJE:

1. void dodajClan(int stepen, int koeficijent) – ukoliko je koeficijent jednak nuli, funkcija ignoriše takav član. U suprotnom funkcija provjerava da li je proslijeđeni stepen veći od prvog stepena, te ako jeste poziva funkciju dodajNaPocetak koja pravi novi Clan te podešava pokazivače heada i trenutnog prvog člana. Ukoliko je proslijeđeni stepen manji od stepena posljednjeg elementa funkcija poziva funkciju dodajNaKraj koja kreira novi član te podešava pokazivače taila i trenutnog posljednjeg elementa. Ako ipak nije ispunjen nijedan od ova dva uvjeta, funkcija ide iterataorom kroz polinom dok ne naiđe do člana čiji je stepen manji od proslijeđenog stepena, te pravi novi clan, podešava pokazivače i povećava broj članova.
2. void obrisiClan(int stepen) – ova funkcija prvo provjerava da li je član, čiji je stepen jednak proslijeđenom stepenu, prvi ili posljednji (jer u tom slučaju štedimo na vremenu) te briše pokazivač na taj član i podešava pokazivanje njegovog sljedećeg i prethodnog člana. U suprotnom funkcija ide kroz polinom dok ne naiđe do člana čiji je stepen jednak proslijeđenom stepenu te taj član briše i podešava pokazivače njegovog prethodnog i sljedećeg člana. Ukoliko funkcija ne naiđe na član sa proslijeđenim stepenom, baca se izuzetak.
3. friend const Polinom operator+(const Polinom& prvi, const Polinom& drugi) – funkcija prvo provjerava da li jedan od polinoma nema članova, te ako je to slučaj vraća kao rezultat drugi polinom. Ukoliko su oba polinoma nenulti, onda im poredi stepene krećući se od prvog člana ka posljednjem. Ukoliko su im stepeni isti, onda sabira ta dva člana tako što im sabere koeficijente i ostavi isti stepen te taj član dodaje u rezultat. Ako su stepeni različiti u rezultat dodaje član sa većim stepenom te prelazi na sljedeći član polinoma u kojem je naišao na veći stepen. Ovaj postupak se ponavlja sve dok se ne dođe do kraja jednog polinoma. Kada se to desi onda funkcija dodaje preostale članove drugog polinoma u rezultat.
4. friend const Polinom operator-(const Polinom& prvi, const Polinom& drugi) – funkcija radi slično kao prethodno objašnjeni preklopljeni operator sabiranja. Razlika je u tome što prilikom provjere, ukoliko prvi polinom nema članova, funkcija kao rezultat vraća drugi polinom pomnožen sa -1. Druga razlika je u tome što kada funkcija naiđe na članove sa istim stepenom, njihove koeficijente oduzima i ostavlja njihov stepen, te taj član dodaje u rezultat, a kada dodaje članove drugog polinoma njihove koeficijente množi sa -1.
5. friend const Polinom operator\*(const Polinom& prvi, const Polinom& drugi) – funckija je implementirana tako da pomoću dvije for petlje ide kroz prvi polinom i svaki njegov član množi sa svakim članom drugog polinoma.
6. friend const Polinom operator^(const Polinom& prvi, int stepen) – ovo je rekurzivna funckija koja staje ukoliko je stepen jednak 0 ili 1. Za nulu vraća kao rezultat polinom koji se sastoji samo od člana sa koeficijentom 1 i stepenom 0, a za 1 vraća polinom nad kojim je operator pozvan. U suprotnom vraća proizvod dva polinoma koja dobijemo tako što stepenujemo polinom nad kojim je funkcija prvobitno pozvana stepen/2 puta ili (stepen + 1)/2 puta u zavisnosti od toga da li je prvobitno proslijeđeni stepen bio paran ili neparan.
7. friend istream& operator>>(istream& Upis, Polinom& pol) – operator upisa radi slično kao parser koji je rađen na vježbama. Definišemo stack polinoma i stack operacija, te pri unosu broja ili znaka x te članove pretvorimo u polinome i guramo ih na stack polinoma. Kada unesemo +, -, \*, ^, ( te znakove guramo na stack operacija. Pri unosu znaka ')' izvršavamo sve operacije koje su unesene nakon znaka '(' nad polinomima na stacku polinoma. Na kraju na stacku polinoma bi trebao biti samo jedan polinom koji je zapravo traženi oblik unesenog polinoma. Pri implementaciji ove funkcije korištene su dodatne funkcije nečlanice klase Polinom: Prioritet(char operacija) koja nam vraća koliki je prioritet određene operacije da bismo znali šta se prije treba izvršavati i funkcija Izvrsi\_Operaciju(stack<Polinom> &s, char operacija) koja izvršava proslijeđenu operaciju nad posljednja dva polinoma u proslijeđenom stacku.
8. friend ostream& operator<<(ostream& Ispis, const Polinom& pol) – preklopljeni operator ispisa ispisuje 0 ukoliko polinom nema članova. U suprotnom prvo ispiše prvi član u zavisnoti od toga kakav mu je koeficijent. Zatim ide kroz ostale članove sve do posljednjeg i ispisuje ih tako što prvo ispiše predznak njihovog koeficijenta (za šta sam koristila funkciju nečlanicu klase Polinom: Znak(int a)) a zatim apsolutnu vrijednost koeficijenta. Posljednji član se naknadno ispisuje pošto se pri unosu i pri dodavanju članova polinom konstruiše tako da članovi sa manjim stepenima idu na kraj. Zbog toga slobodni član može biti samo posljednji član, i upravo zbog toga da se uz koeficijent ne bi ispisivalo x^0 za ovaj član moramo provjeriti kakav mu je stepen pa tek onda ga ispisati. Takođe u svakom ispisu se provjerava da li je koeficijent uz neki član 1 ili -1 te ako jeste samo se ispisuje znak tog koeficijenta a zatim x^stepen.