

Projekat - Matematika 4

Branko Kokanović 02/067

Maj, 2004

Sadržaj

1	Uvod	3
1.1	Tekst projekta	3
1.2	Osnovna ideja	3
2	Program	4
2.1	JavaScript deo	4
2.2	Java applet deo	5
2.2.1	init metoda	5
2.2.2	paint metoda	6
2.2.3	Ostalo	6
3	Zaključak	7
3.1	Prednosti i mane ovog pristupa	7
3.2	Poznate greške	7
3.3	Korišćeni softver	7
3.4	Korišćena literatura	7

1 Uvod

Za projekat je odabrana grupa A i prvi zadatak u njoj.

1.1 Tekst projekta

Merenjem su snimljeni podaci u N tačaka $(x_i, y_i), (i = 1, 2, \dots, N)$. Da bi se preciznije nacrtao grafik podataka, između svake dve susedne tačke treba umetnuti K tačaka. Korišćenjem interpolacionog polinoma po izboru, odrediti umetnute tačke i nacrtati oba grafika. Zadatak uraditi na dva različita načina i uporediti rezultate.

(Zadati tabelu koordinata, recimo generisanu funkcijom $e^{ax} \cos(bx + c)$, broj tačaka N i broj umetnutih tačaka K).

1.2 Osnovna ideja

Osnovna ideja je bila da se koristi Java za izradu projekta. Međutim, najveći problem je bio unošenje podataka i crtanje grafika. Vagajući jednostavnost realizacije oba problema na različitim platformama i načinima korišćenja koje Java nudi, odlučeno je da se koristi jedna vrsta simbioze Java skripta i apleta koja će biti objašnjena u sledećem poglavlju.

2 Program

Sam program je sastavljen iz dva dela.

- JavaScript
- Java applet

2.1 JavaScript deo

JavaScript deo vrši ulogu korisničkog interfejsa koji služi za prikupljanje ulaznih podataka od korisnika i njihovo prosleđivanje apletu. Sama strana se sastoji od formulara u kome korisnik unosi date tačke ili odabira da se, prema uslovu zadatka one generišu odgovarajućom funkcijom (ukoliko se koristi drugi način, korisnik unosi parametre a, b, c date funkcije, opseg generisanja, kao i broj tačaka N). Korisnik mora uneti i K - broj tačaka za umetanje, kao i način gde će se nalaziti ove tačke. Ostale opcije su vezane isključivo za preglednost i lakoću korišćenja i, u principu, nemaju veze sa tekstom zadatka. Sam JavaScript omogućuje unošenje već određenih vrednosti, tako da se, po startovanju Internet strane, odmah može ići na crtanje grafika.

Po učitavanju apleta, poziva se funkcija `Ucitaj()` koja

- Ukazuje na moguće greške i zaustavlja učitavanje ukoliko nije unet neki traženi parametar,
- Ukoliko je korisnik odabrao da se tačke N generišu, generiše ih funkcijom `GenerisiTacke(N,a,b,c,min,max)`
- Pravi novu dinamičku stranicu,
- U novonapravljenu stranu upisuje applet tag,
- U zavisnosti od opcija koje je korisnik zadao, upisuje i parametre koji se prosleđuju apletu i
- Zatvara dinamičku stranicu

čime se otvara nova web strana i počinje izvršavanje apleta.

U ovom odeljku dajemo i izvorni kod funkcije `GenerisiTacke(N,a,b,c,min,max)` (koja vraća tekst koji se koristi u apletu, oblika korišćenog kao i ako korisnik sam unosi tačke)

```
function GenerisiTacke(N,a,b,c,min,max){
    var TackeN="";
    var i;
    var x;
    TackeN="("+min+", "+FunkcijaGenerisanja(min,a,b,c)+")";
    for (i=1;i<N-1;i++){
        x=min+((max-min)*i)/(N-1);
        TackeN=TackeN+"(+x+", "+FunkcijaGenerisanja(x,a,b,c)+")";
    }
}
```

```

        TackeN=TackeN+"("+max+", "+FunkcijaGenerisanja(max,a,b,c)+")";
        return(TackeN);
    }

    function FunkcijaGenerisanja(x,a,b,c){
        return (Math.exp(a*x)*Math.cos(b*x+c));
    }

```

2.2 Java applet deo

Java applet deo vrši crtanje, ali pored toga, vrši i izračunavanje LAGRANGEovog polinoma. Applet je relativno jednostavan i uglavnom sav posao vrše nasleđene metode Applet klase `init` i `paint`.

2.2.1 init metoda

Funkcija `init` vrši:

- Učitavanje parametara prosleđenih od JavaScript-a
- Učitavanje N tačaka u niz parsovanjem prosleđenog teksta
- Računanje K tačaka LAGRANGEovim interpolacionim polinomom
- Konfigurisanje grafika

Prve dve stvari su rutinske, dok je nama zanimljiva treća. Za date tačke (x_i, y_i) , $i = 1, 2, \dots, N$, LAGRANGEovim interpolacioni polinom glasi:

$$L(x) = \sum_{i=1}^N y_i \prod_{\substack{j=1 \\ j \neq i}}^N \frac{(x - x_j)}{(x_i - x_j)}.$$

Odlučeno je da funkcija `Lagrange` dobija prosleđenu vrednost prve koordinate, a da vraća `y`. Ovo se može uraditi, a i urađeno je na sledeći način:

```

private float Lagrange(float x){
    int i, j;
    float Lagrange, LagrangeTemp;
    Lagrange=0;
    for (i=0; i<N; i++){
        LagrangeTemp=1;
        for (j=0; j<N; j++){
            if (i!=j){
                LagrangeTemp=LagrangeTemp*(x-TackeN[j].x());
                LagrangeTemp=LagrangeTemp/(TackeN[i].x()-TackeN[j].x());
            }
        }
        LagrangeTemp=LagrangeTemp*TackeN[i].y();
        Lagrange=Lagrange+LagrangeTemp;
    }
    return Lagrange;
}

```

Konfigurisanje grafika je zapravo pronalaženje maksimalne apsolutne vrednosti svih tačaka za obe koordinate.

2.2.2 paint metoda

Ova metoda vrši iscrtavanje grafika. Olakšavajući posao programiranja je onemogućio veću fleksibilnost pri iscrtavanju, tako da grafik uvek ima $(0,0)$ na sredini apleta, bez obzira na tačke. Prvo se iscrtava kostur apleta i upisuju maksimalne vrednosti, nađene u `init` metodu. Maksimalne vrednosti su podešene na 95% širine i visine tako da uvek znamo gde je koja tačka i to nam je referenca za crtanje ostalih tačaka.

2.2.3 Ostalo

Dodato je da se prati miš i ispisuje vrednost tačke iznad koje se nalazi u *status bar*-u korišćenog *browser*-a. Način pronalaženja tačke je inverzan načinu crtanja tačke pomenutom gore.

Samo jedna korisnička klasa je definisana - **Tacka** zato što Java ima klasu **Point** kojoj su koordinate celi brojevi, a nama je bio potreban rad sa **float** veličinama.

3 Zaključak

3.1 Prednosti i mane ovog pristupa

Prednosti programiranja u Javi, kao i ovog programa su:

- Nezavisnost od operativnog sistema
- Objektno orijentisano programiranje i sve što ono donosi

dok na ova dva velika plusa dolazi i nekoliko manjih i ispravljivih minusa:

- Nepodržanost *java-enabled browser*-a kod slabije upućenih korisnika
- Slaba obrada grešaka
- Nemogućnost zumiranja grafika

3.2 Poznate greške

- Na *Netscape Navigator*-u (Linux) se iscrtavaju N tačke, linije između njih, K tačke, dok se linije za njih ne iscrtavaju pravilno, već im je kraj pomeren na y osi, iako je korišćen isti kod.

3.3 Korišćeni softver

1. Notepad (razvojno okruženje, java, javascript)
2. **Eclipse Platform v 2.1.2**(razvojno okruženje, java)
3. **L^AT_EX** (dokumentacija)

3.4 Korišćena literatura

1. Dobrilo Đ. Tošić, Mirko S. Jovanović, Branko J. Malešević, *Ispitni zadaci iz Matematike IV,III* izdanje, AKADEMSKA MISAO, Beograd, 2002.
2. JavaTM 2 Platform, v 1.3 API Specification