

Projekt POLINOMI

Naročnik: Komparator d.d.
Vodja projekta : Janez Ferin

Status dokumentov

- [*] D1. Naročnikove zahteve
- [*] D2. Plan projekta
- [*] D3. Sistemske specifikacije
- [*] D4. Testni primeri
- [*] D5. Poročilo o preverjanju
- [*] D6. Načrtovalska dokumentacija
- [*] D7. Uporabniški priročnik
- [*] D8. Izvorna koda – izpis
- [*] D9. Izvršilna koda
- [*] D10. Opis vseh datotek

Začetek: 1.3.2002

Konec: 8.4.2002

Ime in priimek	Vloga	Naslov	Opomba
Janez Ferin	Preverjanje	ferin@uni-mb.si	
Miha Štukovnik	Razvoj	miha@uni-mb.si	

Rubrika, ki je namenjena pregledovalcu projekta

Projekt oddan:

Ta zgled je namenjen študentom, ki poslušajo predmet Osnove programskega inženirstva na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru.

Razvoj kompleksne in kakovostne programske opreme mora potekati po nekem ustaljenem vzorcu, ki običajno temelji na določenem standardu. Za potrebe študentov sem sestavil preprost standard, na katerem temelji ta zgled. Obširnost dokumentacije je odvisna predvsem od kompleksnosti programa in zahtevane kakovosti. Za zgled sem izbral preprost program, ki zagotovo ne opravičuje tako obsežne dokumentacije. Pri izboru sem imel v mislih predvsem to, da program v zgledu ne sme biti prezahteven in predolg, sicer s tem zgledom ne bi dosegel svojega namena - to je ilustracija razvoja kompleksne oziroma zahtevne programske opreme.

Zgled je urejen z MS Word urejevalnikom tako, da je mogoče večino besedila ponovno uporabiti.

Maribor 29.9.2010

Tomaž Dogša

Vse pravice so zadržane.

D1 NAROČNIKOVE ZAHTEVE

Dokument	Naročnikove zahteve, Verzija 1.0, 28.2.2002
Naročnik	Laboratorij za elektronske sisteme
Lokacija dokumenta	datoteka polinomi.doc
Odgovorna oseba	Direktor firme Komparator, J. Globočnik

Spoštovani,

Iščemo ponudnika za izdelavo programske opreme, ki bo v nadaljevanju podrobneje opisana. Najkasneje do 10.3.2002 mi pošljite vaš terminski plan, ki vključuje tudi ceno ter sistemske specifikacije.

Že dlje časa opažamo, da je potrebna računalniška podpora za nekatere načrtovalske aktivnosti. Ena izmed njih se nanaša na izračun ničel polinomov. Na zadnji seji smo se odločili za nakup oziroma izdelavo ustreznega programa. Ugotovili smo, da potrebujemo hiter in kakovosten program, ki bo omogočal izračun ničel polinomov in preprost izris na ekran. Do sedaj so tehniki to nalogo opravljali s pomočjo kalkulatorja. Ta proces iskanja lastnosti polinoma bi radi pospešili in poenostavili. Vnos potrebnih podatkov naj bo preko tastature ali datoteke.

Želeli bi, da program teče na PC računalniku v DOS operacijskem sistemu. Za izvajanje programa mora zadoščati standarden PC. Program naj bo robusten in prijazen. Brezpogojno zahtevamo, da program ne vsebuje nobenih napak.

Program in dokumentacija morata biti izdelana in preverjena v skladu s standardom cV&Vs.

Maribor 28.2.2002

Direktor firme Komparator

J. Globočnik

D2 PLAN PROJEKTA

D2.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument: Plan Projekta	
verzija	1
datum	8.4.2002
lokacija	datoteka Projekt_Polinomi.doc v podmapi »dokumentacija« mape projekta
Odgovorna oseba	Janez Ferin
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve

D2.2 KRATEK OPIS PROBLEMA

Firma Komparator d.d. (v nadaljevanju naročnik) je 28.2.2002 predložila svoje zahteve, na podlagi katerih je bil izdelan ta plan projekta.

Naročnik se pri svojem delu srečuje s problemom iskanja ničel polinomov. Potrebuje program (v nadaljevanju Polinomi), ki mu bo omogočal poenostavitev oziroma avtomatizacijo tega dela. Poleg izračuna ničel želi tudi izris grafa polinoma. Program mora omogočati interaktivni vnos polinoma, pa tudi branje iz datoteke.

D2.2.1 Globalni cilji (globalne zahteve), ki jih želimo s produktom doseči

- 1) Program mora zahtevane izračune opraviti bistveno prej kot tehnik z »navadnim« kalkulatorjem.
- 2) Program naj bo dovolj preprost in intuitiven, da ne bo potrebno posebno usposabljanje operaterjev.

D2.2.2 Omejitve (operacijski sistem, aparturna oprema, standardi...)

- 1) Program mora teči na PC računalniku v operacijskem sistemu DOS oziroma v ukaznem oknu znotraj Windowsov.
- 2) Za izvajanje programa Polinomi mora zadoščati standardni PC. Program mora biti dokumentiran skladno s standardom CVVS 2/2000.

D2.2.3 Rok za zaključitev projekta, skupni stroški

- 1) 15.4.2002
- 2) Maksimalni skupni stroški izdelave projekta so 1400 EUR.

D2.2.4 Funkcije

Bistvene funkcije, ki jih mora sistem izvajati, da bodo doseženi globalni cilji:

- 1) Omogočati mora vnos polinoma preko tipkovnice ali branje iz datoteke
- 2) Omogočati mora fleksibilen izris grafa polinoma
- 3) Program mora na 7 decimalnih mest natančno izračunati ničle polinoma
- 4) Navodila morajo biti dostopna med uporabo programa

D2.2.5 Pomembne karakteristike

- 1) Za izračun na srednje zmogljivem računalniku (procesor 500 MHz) program ne sme porabiti več kot 5 sekund. (S tem bomo po naši presoji zadovoljili nepreverljivi subjektivni zahtevi po hitrosti)
- 2) Program mora napačne ali nesmiselne vhodne podatke zavrniti in nadaljevati z delom, kot je najboljše možno. (S tem bomo po naši presoji zadovoljili nepreverljivi subjektivni zahtevi po robustnosti.)

D2.2.6 Neizvedljive zahteve

Ker je popolno odsotnost napak nemogoče zagotoviti, bo ta zahteva zavrnjena.

D2.2.7 Označevanje verzij : V x.y DDMMLLL

X glavna oznaka (velike spremembe), y za majhne spremembe, DDMMLLL : datum)

D2.3 ZAGOTAVLJANJE KAKOVOSTI (NAČRT PREVERJANJA)

D2.3.1 Objekti preverjanja

D1	Naročnikove zahteve
D2	Plan projekta
D3	Sistemske specifikacije, Prototip
D4	Testne primere
D5	Poročilo o preverjanju
D6	Načrtovalsko dokumentacijo
D7	Uporabniški priročnik
D8	Izvorna koda
D9	Izvršljiv program
D10	Kompleten produkt

Glede na izbran model razvoja obstajajo delni in končni produkti, ki jih je potrebno na koncu vsake faze preveriti (glej tabelo Pregled po produktih in aktivnostih). Kompleten terminski plan je podan v 7. točki tega dokumenta. Končni produkt predstavljajo dokumenti D1-D10.

D2.3.2 Uporabljene preverjevalne metode

Vse metode so navedene v tabeli Pregled po produktih in aktivnostih. Dodatna pojasnila:

A) Splošni pregled vmesnih dokumentov (produkti D1 do D7, D10), ki niso programi

Preverjevalec bo osebno pregledal dokument in sporočil odgovorni osebi vse ugotovljene nepravilnosti. O preverjanju ne bo nobenega posebnega poročila razen v primeru večjega števila neustreznosti.

Preverjala se bo:

- popolnost
- konsistentnost s predhodnimi dokumenti
- skladnost dokumenta s standardom CVVS-2/2000.

B) Evalvacija prototipa

Prototip bomo preverili s pregledom izvirne kode (stil kodiranja, skladnost s standardom) in testiranjem. Posebej za evalvacijo bodo pripravljene določeni testni vzorci in postopki, ki jih bo natančneje definirala dokument Testni primeri. Evalvacijo izvaja preverjevalec, avtor je prisoten. Po evalvaciji se napravi kratek interni zapisnik. Na podlagi zapisnika se izvede odpravljanje neustreznosti. Ne izvaja se nobenih regresijskih testov.

C) Preverjanje končnega produkta

Preverjanje izvirne kode:

- S splošnim pregledom bomo preverili skladnost kodiranja gleda na standard cV&Vs.

Preverjanje izvršljivega programa:

Na podlagi tega dokumenta (Plan projekta) bo izdelan dokument z naslovom Testni primeri. Med testiranjem se bo vodil osebni zapisnik preverjevalca (uporabi se lahko vnaprej pripravljen formular). Pogoji za začetek testiranja je, da tipični testni vzorec ne sme povzročiti resne hibe.

Uporabljene bodo naslednje strategije (podroben opis je v prilogi tega dokumenta):

- a) prisotnost zahtev (Z)
- b) prepovedane vrednosti – za preverjanje robustnosti (R)
- c) enakovredni razredi- samo za polinom (E)

- d) mejne vrednosti (M)
- e) ugibanje napak oziroma nepravilnosti (U)

D2.3.3 Odpovedna kriterijska funkcija

Izračunani rezultati se morajo na 7 decimalnih mest (plavajoča vejica) natančno ujemati z dejanskimi.

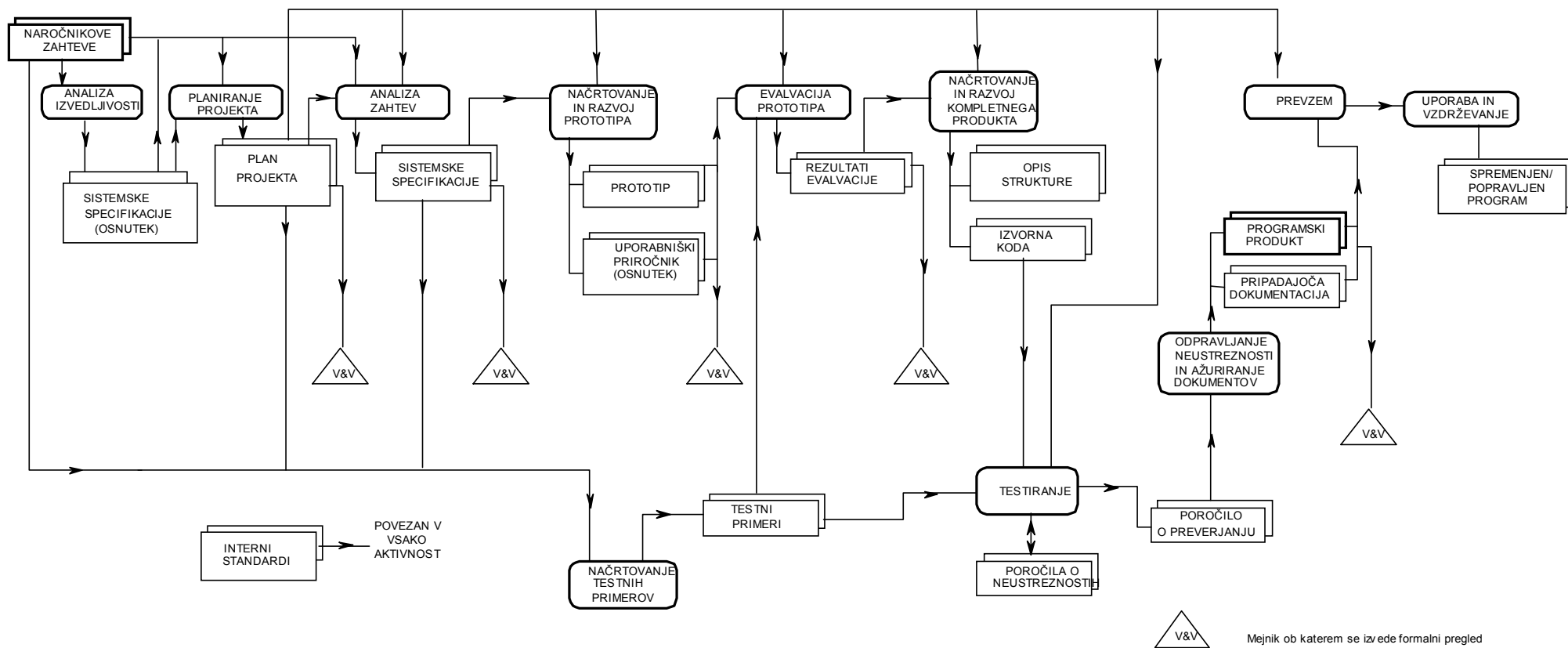
D2.3.4 Terminalna kriterijska funkcija

S testiranjem končamo, ko je izpolnjen pogoj: (pg1 IN pg2) ALI pg3.

- pg1. Preveriti je potrebno prisotnost vseh zahtev, ki so podane v sistemskih specifikacijah.
- pg2. Vsaka funkcija v izvorni kodi mora biti klicana najmanj enkrat.
- pg3. Ko preteče predvideno obdobje, ki je namenjeno testiranju.

D2.4 NALOGE IN REZULTIRAJOČI DOKUMENTI (IZBRAN RAZVOJNI MODEL)

Izbrali smo razvojni model določen s standardom CVVS 2/2000.



Slika 1 - Izbran model razvoja programske opreme (iz vsake faze se lahko vrnemo na predhodne faze - zaradi preglednosti ni narisano)

Pregled po produktih in aktivnostih

	Produkt	Planirana kompleksnost ¹	Dejanska kompleksnost ¹	Odgovorna oseba za produkt	V&V metoda	Odgovorna oseba za V&V	Način sporočanja o V&V	Opomba
D1	Naročnikove zahteve	1 stran	1 stran	J. Ferin	splošni pregled		ustno	
D2	Plan projekta	5 strani	10 strani	J. Ferin	splošni pregled		ustno	
D3	Sistemske speci.	10 strani	8 strani	J. Ferin + naročnik	splošni pregled	J. Ferin	ustno	
	Prototip	500 LOC	575 LOC	M. Štukovnik	testiranje+ splošni pregled	J. Ferin	interni zapisnik	
D4	Testni primeri	20 testnih primerov	21 testnih primerov	J. Ferin	splošni pregled	J. Ferin	ustno	
D5	Testno poročilo	5 strani	17 strani	J. Ferin	splošni pregled	J. Ferin	ustno	
D6	Načrtovalska dokumentacija	3 strani	7 strani	M. Štukovnik	splošni pregled		ustno	
D7	Uporabniški priročnik	3 strani	4 strani	M. Štukovnik	splošni pregled		ustno	
D8	Izvirna koda	2500LOC	1257LOC	M. Štukovnik	splošni pregled	J. Ferin	ustno	
D9	Izvršljiv program	300KB	72KB	M. Štukovnik	testiranje	J. Ferin	pisno	Poročilo o preverjanju
	Kompleten produkt			vsi	testiranje+ splošni pregled	Naročnik		

Roki in stroški

	AKTIVNOST	Planiran rok	Dejanski rok	Planirani napor ¹	Planirani stroški	Dejanski napor	Dejanski stroški	Izvajalec	Odgovorna oseba
A1	Planiranje projekta in analiza zahtev	7.3.2002	6.3.2002	5	250	4	200	J. Ferin	J. Ferin
A2	Načrtovanje	12.3.2002	8.3.2002	3	150	2	100	M. Štukovnik	J. Ferin
A3	Implementacija prototipa	15.3.2002	12.3.2002	3	150	2	100	M. Štukovnik	J. Ferin
A4	Evalvacija prototipa	18.3.2002	13.3.2002	1	50	1	50	J. Ferin	J. Ferin
A5	Implementacija programa	25.3.2002	22.3.2002	5	250	7	350	M. Štukovnik	J. Ferin
A6	Načrtovanje testnih primerov	20.3.2002	18.3.2002	2	100	3	150	J. Ferin	J. Ferin
A7	Testiranje in popravljanje kompletnega programa	29.3.2002	29.3.2002	4	200	5	250	M. Štukovnik	J. Ferin
A8	Izdelava kompletne dokumentacije	3.4.2002	5.4.2002	3	150	5	250	M. Štukovnik	J. Ferin
A9	Pregled izvirne kode	4.4.2002	8.4.2002	1	50	1	50	Naročnik + J. Ferin	J. Ferin
A10	Prevzem	4.4.2002	8.4.2002	1	50	1	50	Naročnik + J. Ferin	J. Ferin
	Skupaj napor - stroški			28	1400	31	1550		

Enota napora: človek-dan
Stroški enote napora: 50 EUR

Povzetek stroškov in ocena dobička:

	planirano	planirano (%)	dejansko	dejansko (%)
stroški – analiza in načrtovanje (A1+A2)	400	29%	300	21%
stroški - kodiranje in izdelava dokumentacije (A3+A5+A8)	550	39%	700	50%
stroški - preverjanje (A4+A6+A7+A9)	400	29%	500	36%
stroški – ostalo	50	3%	50	3%
stroški – skupaj	1400	100%	1550	110%
dobiček	600	30%	450	23%
Naročnik bo glede na pogodbo plačal	2000	100%	2000	100%

¹ Navedi tudi ustrezno enoto (npr.: enota so ure ali stroški ali ljudje ali kombinacija).

D2.5 RESURSI

D2.5.1 Osebe (Kdo bo sodeloval, kakšna je njegova vloga, kakšne morajo biti njegove sposobnosti?)

	Oseba	AKTIVNOST	Vloga
P1	J. Globočnik	<ul style="list-style-type: none">prevzemnadzor	naročnik
P2	Janez Ferin	<ul style="list-style-type: none">planiranje projektaanaliza zahtevnačrtovanje testnih primerovprevzem, Nadzor	preverjevalec (vodja projekta)
P3	Miha Štukovnik	<ul style="list-style-type: none">načrtovanjeimplementacija prototipaevalvacija prototipanačrtovanje in implementacija kompletnega programapopravljanje kompletnega programaizdelava uporabniškega priročnika	razvojniki

D2.5.2 Potrebna programska orodja, knjižnice

orodje	namen, funkcija
Microsoft Visual C++ 6	kodiranje, odpravljanje neustreznosti
Microsoft Word	vodenje dokumentacije
Winmetr	merilnik kompleksnosti
Instr	instrumentator

D2.5.3 Potrebna aparatura oprema

orodje	namen, funkcija
dva računalnika PC	kodiranje, odpravljanje neustreznosti, vodenje dokumentacije
tiskalnik	izpis dokumentacije

D2.6 RAZDELITEV STROŠKOV

Glej točko D2.4.

D2.7 TERMINSKI PLAN PROJEKTA

Časovne enote so (minute,ure,dnevi): (delovni) dnevi

1 dan: 1. marec 2002

26 dan: 8. april 2002

AKTIVNOST		ČASOVNA SKALA																														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
A1	Planiranje projekta in analiza zahtev	+	+	+	+	+																										
A2	Načrtovanje					*	+	+	+																							
A3	Implementacija prototipa							*	*	+	+	+																				
A4	Evalvacija prototipa									*			+																			
A5	Implementacija kompletnega programa										*	*	*	+	+	+	+	+														
A6	Načrtovanje testnih primerov										*	*	*	+	+																	
A7	Testiranje in popravljanje kompletnega programa																	*	+	+	+	+	+									
A8	Izdelava kompletne dokumentacije																					+	+	+	*	*						
A9	Pregled izvorne kode									+	*											+	*									
A10	Prevzem																								+		*					
	DOKUMENT (skrajni rok)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
D1	Naročnikove zahteve	+	*																													
D2	Plan projekta			+	*																											
D3	Sistemske specifikacije						*		+																							
D4	Testni vzorci												*		+																	
D5	Testno poročilo																					+	*									
D6	Načrtovalska dokumentacija																															
D7	Uporabniški priročnik																							+		*						
D8	Izvorna koda																*	+														
D9	Izvršljiv program																					+	*									

Legenda:

Trajanje aktivnosti: + planiran čas, * dejansko porabljen čas

D2.8 POJMOVNIK

pojmem	razlaga
naročnik	Komparator d.d.

D2.9 PRILOGE

D2.9.1 Opisi uporabljenih strategij

1. Opis strategije: Prisotnost zahtev (Z)

1. Strategija je uporabna je v vseh primerih, kjer so znane specifikacije oziroma zahteve, med katerimi ni nobenih relacij. Uporablja se lahko za testiranje kompletnega programa, ali pa samo komponente
2. Predpostavka o napaki: določena zahteva ni implementirana. S to strategijo odkrivamo zahteve, ki niso implementirane. Razen zelo redkih izjem, ne bomo odkrili napačno implementiranih zahtev in zahtev, ki so po nepotrebnem implementirane.
3. Testirni model je seznam zahtev.
4. **Pravilo za načrtovanje testnih primerov:** Za vsako zahtevo tvori najmanj en testni primer. Vhodne podatke si poljubno izberi.
5. Z načrtovanjem testnih primerov lahko začnemo, ko so zahteve postavljene.
6. Testirna strategija je izčrpana, ko preverimo prisotnost vsake zahteve v seznamu.

2. Opis strategije za preverjanje robustnosti (R)

1. Strategija je uporabna je v vseh primerih, kjer je zahtevana robustnost in je možno tvoriti opis vhodne domene.
2. Predpostavka o nepravilnosti: program ni robusten, čeprav bi moral biti. S to strategijo ne bomo odkrili nepravilnosti, ki se pojavljajo pri procesiranju veljavnih podatkov.
3. Testirni model je opis vhodne domene.
4. **Pravilo za načrtovanje testnih primerov:** V vhodni domeni in identificiraj prepovedane razrede. Za vsak prepovedan razred tvori en testni primer.
5. Z načrtovanjem testnih primerov lahko začnemo, ko je opisana vhodna domena.
6. Testirna strategija je izčrpana, ko smo pokrili vse neveljavne razrede v vhodni domeni. Zgornje število testnih primerov je enako številu neveljavnih razredov.

3. Opis strategije: Ekvivalentni razredi polinomov (E)

1. Strategija je uporabna je v vseh primerih, kjer so znana pravila za opis polinoma.
2. Predpostavka o nepravilnosti: določen razred se procesira nepravilno.
3. Testirni model je opis vhodne domene, ki ustreza lastnostim polinoma.
4. **Pravilo za načrtovanje testnih primerov:** Identificiraj enakovredne razrede, glede na skupno lastnost. Uporabi samo veljavne razrede. Za vsak enakovredni razred tvori en testni primer.
5. Z načrtovanjem testnih primerov lahko začnemo, ko je opisana vhodna domena.
6. Testirna strategija je izčrpana, ko smo pokrili vse enakovredne razrede v vhodni domeni. Zgornje število testnih primerov je enako številu enakovrednih razredov.

4. Opis strategije: uganjanje nepravilnosti (U)

1. Strategija je splošno uporabna.
2. Predpostavlja se, da je prisotna določena nepravilnost ali napaka.
3. Testirni model je seznam potencialnih nepravilnosti oziroma napak.
4. **Pravilo za načrtovanje testnih primerov:** Za vsako potencialno napako oziroma nepravilnost v seznamu tvorimo en testni primer, s katerim preverimo, ali je ta napaka/nepravilnost prisotna.
5. Z načrtovanjem testnih primerov lahko začnemo, ko je imamo pripravljen seznam.

6. Testirna strategija je izčrpana, ko smo pokrili celoten seznam. Zgornje število testnih primerov je enako številu napak oziroma nepravilnosti v seznamu.

5. Opis strategije: mejne vrednosti (M)

1. Strategija je splošno uporabna.
2. Predpostavka o nepravilnosti: vhodni podatki, ki se nahajajo v okolici ali pa točno na meji med veljavnim in neveljavnim območjem, se bodo nepravilno procesirali.
3. Testirni model je vhodna in izhodna domena.
4. **Pravilo za načrtovanje testnih primerov:** določi meje med veljavnimi in neveljavnimi podatki. Izberi vrednost točno na meji, malo nad in malo pod njo.
5. Z načrtovanjem testnih primerov lahko začnemo, ko je imamo podatkovni slovar.
6. Testirna strategija je izčrpana, ko smo uporabili vse meje.

6. Opis strategije: strukturno testiranje¹ (S)

1. Strategija je uporabna predvsem za testiranje komponent. Možno pa jo je uporabiti tudi za testiranje celotnega programa. Najprej instrumentiran program testiramo s testnimi primeri, ki smo jih tvorili na podlagi raznih strategij. Če se pokaže, da določeni deli programa (stavki) niso bili nikoli uporabljeni, potem uporabimo to strategijo.
2. Predpostavka o napaki: posledica napake so bo pojavila, ko se bo uporabil določen del programske kode.
3. Testirni model je graf programa.
4. **Pravilo za načrtovanje testnih primerov:** Ugotovimo, kateri del programske kode še ni bil uporabljen in nato z analizo programa določimo zahtevane vhodne podatke, ki bodo pokrili nepokriti segment.
5. Z načrtovanjem testnih primerov lahko začnemo, ko dobimo podatke o pokritosti stavkov, ki je bila dosežena s predhodnimi testnimi strategijami.
6. Testirna strategija je izčrpana, ko smo pokrili vse programske stavke.

¹ Uporabi le, če je na voljo ustrezna programska podpora.

D3 SISTEMSKE SPECIFIKACIJE

D3.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument: Sistemske specifikacije	
verzija	1
datum	8.3.2002
lokacija	datoteka Projekt_Polinomi.doc v podmapi »dokumentacija« mape projekta
Odgovorna oseba	
	Miha Štukovnik
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve
	Plan projekta, V0.9

D3.2 POVZETEK

Naročnik je 28.2.2001 predložil svoje zahteve, na podlagi katerih so bile izdelane te sistemske specifikacije.

Naročnik se pri svojem delu srečuje s problemom iskanja ničel polinomov. Potrebuje program, ki mu bo omogočal poenostavitev oziroma avtomatizacijo tega dela. Poleg izračuna ničel želi tudi izris grafa polinoma. Program mora omogočati interaktivni vnos polinoma, pa tudi branje iz datoteke. Za iskanje ničel naj zadošča le vnos polinoma v običajni algebrski obliki.

D3.3 ZAHTEVE GLEDE POSAMEZNIH KARAKTERISTIK

Kritične zahteve so označene s klicajem.

D3.3.1 Korektnost oziroma funkcionalnost

Glej točko D 3.7.

D3.3.2 Zanesljivost

Naročnik brezpogojno zahteva, da program ne vsebuje nobenih napak. Tega ni mogoče jamčiti, zato bomo program le podrobno preizkusili v skladu s standardom.

D3.3.3 Testabilnost

Program mora delovati v dveh delovnih režimih: v normalnem in testnem. V testnem režimu naj program izpisuje dodatne diagnostične podatke, denimo številko forme ipd.. Testni režim vključimo s parametrom -t na ukazni vrstici.

D3.3.4 Prenosljivost

Ni zahtevana. Program teče v ukaznem načinu (DOS oziroma ukazno okno znotraj Windowsov) na PC računalniku.

D3.3.5 Prijaznost

Naročnik zahteva prijazen in robusten program. Nobena zahteva ni podana na merljiv način. Izdelali bomo menijsko voden program z vgrajeno pomočjo za vse funkcije. V primeru napačnega oziroma nesmiselnega vnosa naj program deluje dalje in uporabnika opozori na napačen vnos.

D3.3.6 Razumljivost

Ni kvalitativnih niti kvantitativnih zahtev.

D3.3.7 Varnost

Ni kvalitativnih niti kvantitativnih zahtev.

D3.3.8 Vzdrževalnost

Program mora biti modularno zgrajen. Mora biti dokumentiran v skladu s standardom.

D3.3.9 Zmogljivost

Za izračun na srednje zmogljivem računalniku (procesor 500 MHz) program ne sme porabiti več kot 5 sekund.

Program mora na 7 decimalnih mest natančno izračunati ničle polinoma.

D3.4 OMEJITVE IN DRUGE ZAHTEVE

- 1) Za izvajanje programa mora zadoščati standardni PC.
- 2) Zagon programa:

```
polinomi [-t]
```

Kretnica -t požene program v testnem režimu delovanja.

D3.5 OPIS SISTEMA

Opis funkcionalnosti je napravljen s pomočjo tipičnih vzorcev uporabe in diagrama toka podatkov.

D3.5.1 Tipični vzorci uporabe

TVZ_1. Ročni vpis polinoma

1. *Zač. stanje*: prompt, uporabnik pozna polinom
2. *Iniciator*: uporabnik potrebuje izpisan izračun ničel za polinom
3. *Opis dogodkov*:
 - a. Vpis polinoma
 - b. Izberemo območje, znotraj katerega naj išče ničle in zahtevano natančnost
 - c. Program izračuna ničle, izriše graf in izpiše tabelo.
4. *Končno stanje*: izpisan graf, ničle in tabela.

TVZ_2. Polinom je opisan na datoteki

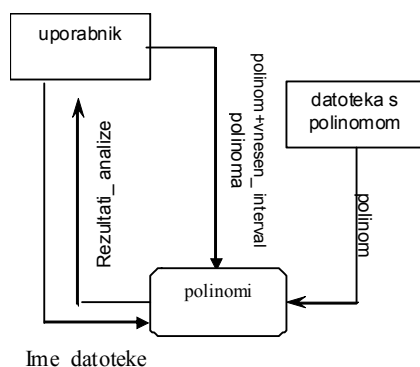
1. *Zač. stanje*: prompt, uporabnik pozna polinom
2. *Iniciator*: uporabnik potrebuje izpisan izračun ničel za polinom
3. *Opis dogodkov*:
 - a. Z navadnim tekstovnim urejevalnikom opišemo polinom in datoteko shranimo
 - b. Poženemo program Polinomi
 - c. Zahtevamo branje iz datoteke
 - d. Nadaljujemo s korakom 3.a. v TVZ_1
4. *Končno stanje*: izpisan graf, ničle in tabela.

TVZ_3. Uporabnik zahteva pomoč

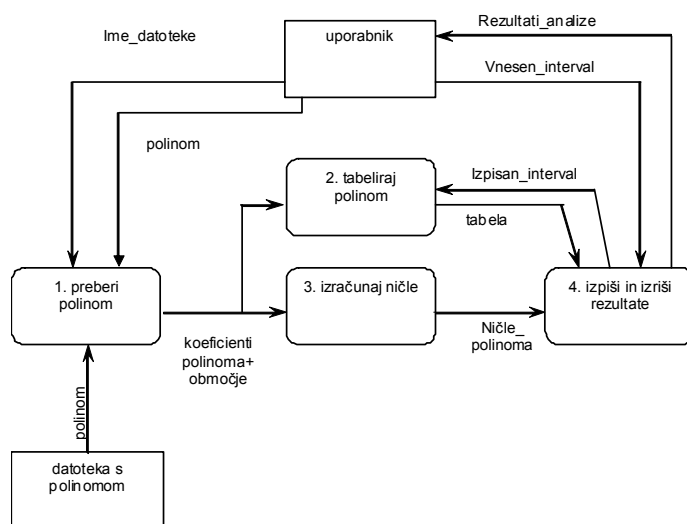
1. *Zač. stanje*: prompt
2. *Iniciator*: uporabnik potrebuje pomoč
3. *Opis dogodkov*:
 - a. Poženemo program Polinomi
 - b. Zahtevamo pomoč.
4. *Končno stanje*: na ekranu je izpisana pomoč.

D3.5.2 Diagram toka podatkov

Ker gre za preprost sistem, smo ga opisali z diagramom toka podatkov.



Slika 2 – Nivo sistema (kontekstni nivo)



Slika 3 - Nivo podsistemov

OPIS PROCESOV

1. preberi polinom

Proces omogoči uporabniku izbiro med interaktivnim vnosom polinoma ali branjem iz datoteke. Če se uporabnik odloči za interaktivni vnos, proces omogoči uporabniku vnos polinoma (*polinom*), sicer pa od njega zahteva ime datoteke (*ime datoteke*), v kateri je polinom zapisan, nakar prebere polinom iz te datoteke (*datoteka s polinomom*).

2. tabeliraj polinom

Proces najprej od procesa *izpiši in izriši rezultate* pridobi podatek o izbranem intervalu tabeliranja (*interval*), od procesa *preberi polinom* pa *koeficiente polinoma*. Nato na tem intervalu izračuna predvideno število vrednosti polinoma pri vrednostih, ki so enakomerno porazdeljene znotraj izbranega intervala.

3. izračunaj ničle !!

S pomočjo izbranega algoritma izračuna *ničle polinoma*.

4. izpiši in izriši rezultate

Proces od uporabnika pridobi *interval* tabeliranja, potem pa od procesa *tabeliraj polinom* pridobi *tabelo* vrednosti polinoma na izbranem intervalu. To tabelo izpiše, hkrati pa jo uporabi za izris grafa polinoma na izbranem intervalu. Od procesa *izračunaj ničle* pridobi seznam ničel in jih izpiše.

D3.6 OPIS PODATKOVNIH TOKOV IN TERMINATORJEV

D3.6.1 Podatkovni slovar za sliko 3

ime podatka	atribut	tip	Veljavno območje (domena)	opomba
polinom=	En ali več koeficientov polinoma	znakovni niz	{"0".."9","x","+","-",".", "^"}	oblika: $5x^4-2x^2+x$
Koeficienti_polinoma=	vrednost+ $x^{\text{eksponent}}$	znakovni niz		Npr. $5.34x^4$
vrednost		znakovni niz	MinReal, MaxReal	$3.12e-3$ (vrednost koeficienta)
eksponent		celo število	0,maxInteger	
Ime_datoteke		znakovni niz		Sintaksa je definirana z vrsto operacijskega sistema
Vnesen_interval=	Vnešena spodnja meja+ Vnešena zgornja meja	realno število realno število	MinReal, MaxReal zgornja meja> spodnja meja	Glej zaslon 4
Izpisani_interval=	Izpisana spodnja meja+ Izpisana zgornja meja	realno število realno število	MinReal, MaxReal zgornja meja> spodnja meja	Glej zaslon 4
tabela=	začetek intervala+ korak+ tabela vrednosti	realno število realno število niz realnih števil	MinReal, MaxReal MinReal, MaxReal MinReal, MaxReal	Glej zaslon 4
Ničle_polinoma=	število ničel+ tabela ničel	celo število niz realnih števil	0 do 10 MinReal, MaxReal	Glej zaslon 4
Rezultati analize=	Tabela+izpisani interval+ničle polinoma+graf			Glej zaslon 4
Graf		znakovni niz		Glej zaslon 4

Konstante, ki so odvisne od prevajalnika: MaxReal = $3.4E38$, MinReal = $-3.4E38$, MaxInteger = 2,147,483,647, MinInteger = -2,147,483,648. Navedene vrednosti veljajo na 32-bitnih prevajalnikih.

D3.6.2 Opis terminatorjev

ime terminatorja	opomba
uporabnik	uporabnik, ki dela s programom polinomi
datoteka s polinomom	tekstovna datoteka, ki vsebuje polinom To je tekstovna datoteka z eno samo vrstico. V njej je zapisan polinom v obliki (primer): $5x^4-2x^2+x+1$

D3.7 PODROBEN OPIS IN INDEKSIRANJE FUNKCIJ IN DRUGIH ZAHTEV, KI JIH JE POTREBNO IMPLEMENTIRATI

Kritične so označene s klicajem.

- F1. Interaktivni vnos polinoma
Omogočen mora biti vnos polinoma preko tipkovnice.
- F2. Branje polinoma z datoteke
Program mora biti sposoben polinom prebrati tudi iz datoteke. Zapis naj bo enak kot v interaktivnem načinu.
- F3. Tabeliranje polinoma
Program mora znati tabelirati polinom na izbranem intervalu z izbranim korakom.
- F4. !! Izračun ničel
- F5. Izpis ničel poljubnega polinoma
- F6. Preprost izris grafa
- F7. Pri vsakem vpisu podatkov mora obstajati na zahtevo izpis pomoči.
- F8. Kontrola vhodnih podatkov (robustnost).
Program mora kontrolirati vhodne podatke in zavrniti neveljavne vnose v meniju, pri vnosu polinomov in mej izračunov ter pri vnosu imena datoteke.
- F9. Izpis verzije ob zagonu programa
Program mora ob zagonu vedno izpisati svojo verzijo.
- Z1. Testni način delovanja.
Program mora podpirati testni režim delovanja, v katerem se izpisujejo dodatne informacije.

D3.8 ZUNANJI VIDEZ

Program je bil pognan v normalnem režimu. V testnem režimu, bi se izpisala v desnem zgornjem kotu še številka zaslona npr. Zaslon 1. Podatki, ki jih je vnesel uporabnik, so zaradi večje preglednosti napisani s poševno pisavo. Pri implementaciji pisava ni poševna.

[illegible]

zaslon 6	Pomoč
<pre> ---[Pomoc]----- Glavni meni programa Polinomi ===== V tem meniju izbiras med tremi možnostmi. 1) Če vpises številko 1 (in pritisnes enter) s tem izberes interaktivni vnos polinoma. Polinom bos lahko vnesel preko tipkovnice. 2) Možnost 2 omogoča vnos imena datoteke, v kateri je zapisan polinom. 3) Izracun nicel in izris grafa funkcije je na voljo le, ko je bil polinom ze uspesno vnesen. Ta izbira sprozi izracun nicel, tabeliranje, izris grafa... 0) Z niclo zakljucimo delo s programom.</pre>	<p>Scenarij uporabe TVZ_1. Na zaslonu 1 je bil izbran izpis pomoči. Podobna pomoč se izpiše ob vsakem vpisu vhodnih podatkov.</p>

D3.9 OPIS FUNKCIJ, KI BODO NAJPREJ IMPLEMENTIRANE V OBLIKI PROTOTIPA

- 1) Glavni meni
- 2) Interaktivni vnos polinoma
- 3) Tabeliranje polinoma
- 4) Izračun ničel polinoma na fiksnem intervalu s fiksno vnaprej določeno natančnostjo (3 decimalke)
- 5) Izris grafa na fiksnem vnaprej določenem intervalu

D3.10 PREVZEMNI KRITERIJI

- 1) Program Polinomi mora biti dokumentiran skladno s standardom CVVS-2/2000.
- 2) Program Polinomi mora biti preverjen na najmanj 10 testnih primerih. Naročnik bo pripravil tri svoje testne primere, ki ne smejo pokazati na prisotnost večjih hib.

D3.11 POJMOVNIK

D4 TESTNI PRIMERI

D4.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument: Testni primeri	
verzija	1
datum	18.3.2002
lokacija	datoteka Projekt_Polinomi.doc in tg25.txt v podmapi »dokumentacija« mape projekta
Odgovorna oseba	
	Janez Ferin
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve
	Plan projekta, V1.0
	Sistemske specifikacije V 1.0
	Uporabniški priročnik V 1.0

D4.2 POVZETEK

Na podlagi naročnikovih zahtev in Plana projekta ter ostalih dokumentov, ki so navedeni v predhodni točki, je bil izdelan ta dokument, ki natančno definira postopek testiranja in testne primere. Dokument je uporaben za evalvacijo prototipa in za testiranje kompletnega programa. Rezultati testiranja so opisani v Poročilu o preverjanju.

D4.3 IDENTIFIKACIJA OBJEKTOV, NA KATERE SE TESTNI VZORCI NANAŠAJO

Testni primeri se nanašajo na program polinomi.exe verzija 1 in 2.

D4.4 OPIS TESTNIH PRIMEROV

Glede na izbrane strategije (glej Plan projekta, točka Zagotavljanje kakovosti) smo tvorili 20 testnih primerov, ki so podrobneje opisani v prilogi. Za izpolnitev pogojev glede pokritosti, je bil tvorjen 1 testni primer. Pregled testnih primerov se nahaja v točki D4.6 (glej testno matriko), opis testnih modelov in primerov je v prilogi. V prilogi je tudi analiza vhodne in izhodne domene, ki jo potrebujemo pri strategiji mejnih vrednosti.

D4.5 OPIS TESTIRNEGA POSTOPKA

D4.5.1 Testiranje programa

V okolju MS Windows (9X/NT) testiramo tako, da najprej odpremo okno z ukaznim odzivnikom (»command prompt«), nakar s pomočjo ukaza CD skočimo v imenik, v katerem je program polinomi.exe, ki ga bomo testirali. Testne primere vpisujemo interaktivno preko tipkovnice. Edini testni primer, ki zahteva zunanjo datoteko, je TP1. Ta zahteva datoteko tp1.txt v podimeniku s projektom.

D4.6 IDENTIFIKACIJA FUNKCIJ IN ZAHTEV TER TESTNA MATRIKA

Koda	Kratek opis	Kom- pleks- nost ¹	Stop- nja kriti- čnos- ti ²	Indikator pomem- bnosti ³	Število testov	T P 1	T P 2	T P 3	T G 4 / 1	T G 4 / 2	T G 4 / 3	T G 4 / 4	T G 4 / 5	T G 4 / 6	T G 4 / 7	T G 5 / 1	T G 5 / 2	T G 5 / 3	T G 6 / 1	T G 6 / 2	T G 6 / 3	T G 7 / 1	T G 7 / 2	T G 7 / 3	T P 8
F1	Interaktivni vnos polinoma	3	3	9 (19%)	15 (21%)	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*				*
F2	Branje polinoma z datoteke	3	2	6 (13%)	5 (7%)	*																*	*	*	
F3	Tabeliranje polinoma	2	2	4 (8%)	10 (14%)	*		*	*	*	*	*	*	*	*										*
F4	Izračun ničel	3	3	9 (19%)	10 (14%)	*		*	*	*	*	*	*	*	*										*
F5	Izpis ničel	1	3	3 (6%)	10 (14%)	*		*	*	*	*	*	*	*	*										*
F6	Izris grafa	3	2	6 (13%)	10 (14%)	*		*	*	*	*	*	*	*	*										*
F7	Pomoč na zahtevo	2	2	4 (8%)	3 (4%)	*	*																		
F8	Kontrola vhodnih podatkov	2	2	4 (8%)	5 (7%)	*													*			*	*	*	
F9	Izpis verzije ob zagonu programa	1	1	1 (2%)	1 (1%)	*																			
Z1	Testni način delovanja	1	2	2 (4%)	2 (3%)	*																			
Z2	Hitrost							*																	
Z3	Natančnost							*																	
Z4	Prijaznost (robustnost)						*									*	*	*	*	*	*	*	*	*	
	Skupaj			48 (100%)	71 (100%)	10	1	7	5	5	5	5	7	7	7	2	2	2	1	3	2	3	3	3	5
	pozitivni, negativni strategija ⁴					PZ	PZ	PZ	NE	NE	NE	NE	NE	NE	NE	NE	NE	NE	NE	NE	NE	NE	NE	NE	PS
	prioriteta pri izvrševanju ⁵					1	1	2	1	1	2	3	1	2	3	x	x	x	3	3	3	1	2	3	k

Opomba: tvori najmanj 10 testnih primerov.

¹ Kompleksnost : izberemo eno izmed metrik npr. LOC, če ni znana uporabimo subjektivno merilo (1 nizka kompleksnost, 2 srednja, 3 visoka).

² Stopnja kritičnosti: 3 zelo kritična, 2 kritična, 1 nepomembna

³ Indikator pomembnosti=število testov= stopnja kritičnosti x kompleksnost

⁴ Strategije: prisotnost zahtev (Z), mejna vrednost (M), enakovredni razredi (E), preverjanje robustnosti (R), ugibanje nepravilnosti (U), strukturno testiranje (S)

⁵ K = na koncu, x: ni pomembno vendar pred strukturnimi testi

D4.7 PRILOGA

D4.7.1 Opis¹ vhodne domene²

	ID	Ime podatka, stanja, dogodka	Tip podatka/Opis domene	Opomba
Polinom	ŠČ	število členov	naravno število VR1=[0,500] NR1=x>500]	
Polinom	KOEF	koeficient	real VR1 = [minReal,maxReal] NR1 = {alfanumerični znaki} NR2 = x> maxReal NR3 = x<minReal	
Polinom	EXP	eksponent	celo število VR1= [0,maxInteger] NR1= x>maxInteger NR2= {alfanumerični znaki}	
ime datoteke, v kateri je opis polinoma	DSK	Disk	En znak VR1= {a:\,b:\,c:\,...z:\, prazno} NR1= {0:\. . 9:\,=\, %:\,...}	
	IMN	Imenik	Niz alfanumeričnih znakov VR1=brez imena VR2=dva podimenika NR1=v imenu je prepovedan znak	glej sintakso za Windows oper. sistem {prazno, ?,*,+,!,...}
	IME	Ime datoteke	Niz alfanumeričnih znakov VR1=ime sestavljeno iz 5 znakov VR2=ime brez pike in ekstenzije NR1= v imenu je prepovedan znak NR2= brez imena, samo pika in ekstenzija	{prazno, ?,*,+,!,...}
interval	SP	spodnja meja	real VR1=[minReal,maxReal] NR1=x> maxReal NR2=x<minReal NR3={alfanumerični znaki}	
interval	ZG	zgornja meja	VR1=[minReal,maxReal] NR1=x> maxReal NR2=x<minReal NR3={alfanumerični znaki}	
natančnost iskanja ničel	NAT	natančnost iskanja ničel	real VR1=[minReal,maxReal]	

¹ Začni s podatki, ki se nahajajo v podatkovnem slovarju.

² Opiši najmanj 10 najpomembnejših elementov (podatek, stanje, dogodek). VR: veljavna domena, NR: neveljavna domena

			NR1=x> maxReaL NR2=x<minReal NR3={alfanumerični znaki}	
Ponovitev risanja z novimi mejami	PON	Ponovitev risanja z novimi mejami	En znak	
Meja narisane grafa	XMIN	Spod. meja	VR1={1,0,*} NR1=lzven VR1 real	
Meja narisane grafa	XMAX	Zg. meja	VR1=[minReal,maxReal] NR1=x> maxReaL NR2=x<minReal NR3={alfanumerični znaki} real	
Meja narisane grafa	YMIN	Spod. meja	VR1=[minReal,maxReal] NR1=x> maxReaL NR2=x<minReal NR3={alfanumerični znaki} real	
Meja narisane grafa	YMAX	Zg. meja	VR1=[minReal,maxReal] NR1=x> maxReaL NR2=x<minReal NR3={alfanumerični znaki} real	
Obstoj datoteke	OBST		VR1=[minReal,maxReal] NR1=x> maxReaL NR2=x<minReal NR3={alfanumerični znaki} boolean	
Opis polinoma na datoteki	POLD		VR1=obstaja NR1=ne obstaja	
Izbor v meniju	IN		VR1=katerikoli polinom v eni vrstici VR2=katerikoli polinom v eni vrstici+prazna vrstica NR1=prazna datoteka NR2=prazna vrstica en znak VR1={1,2,*,0}	

Konstante, ki so odvisne od prevajalnika: MaxReal = 3.4E38, MinReal = -3.4E38, MaxInteger = 2,147,483,647, MinInteger = -2,147,483,648. Navedene vrednosti veljajo na 32-bitnih prevajalnikih.

D4.7.2 Opis izhodne domene

	ID	Ime podatka, stanja, dogodka	Tip podatka/Opis razreda	Opomba
	MINX	začetek intervala	real VR1=[minReal,maxReal] NR1= $x > \maxReal$ NR2= $x < \minReal$	
	MAXX	Konec intervala	real VR1=[minReal,maxReal] NR1= $x > \maxReal$ NR2= $x < \minReal$	
ničle polinoma	ŠN	število ničel	Celo število	
			VR1=[0,MaxInteger] NR1= $x > \maxInteger$	
Narisan graf	GRAF		slika VR1=viden približen potek grafa	
			VR1= VR2= NR1= NR2=	

Ekvivalentni razredi polinomov

	ID	Ime podatka, stanja, dogodka	Tip podatka/Opis razreda	Opomba
Vrsta polinoma	VP			
			VR1= Z eno ničlo VR2= Dve ničli VR3= Več kot 2 ničli VR4 =Neskončno ničel VR5= Brez ničel VR6= Polinom, ki se enkrat dotakne abscise VR7= Polinom, ki se večkrat dotakne abscise NR1=nepravilen opis polinoma	

D4.7.3 Priloga s konkretnimi testnimi primeri

Testni primeri so urejeni po strategijah.

Strategija Preverjanje prisotnosti zahtev (Namen: Ali so implementirane vse zahteve?)

TP1-1 Ali je prisotna najbolj kritična funkcionalnost? Ali je implementiran testni režim?

TP1-2 Ali je vedno prisotna vgrajena pomoč?

TP1-3 Ali izračuna ničle z zahtevano natančnostjo? Ali je dovolj hiter?

Strategija Ekvivalentni razredi polinomov (Namen: Pokriti vso vhodno domeno.)

TG2-1 Ali izračunava ničle za vsak ekvivalentni razred? Ali upošteva spremembo območja risanja?

TG2-1/1 Polinom z 1 ničlo.

TG2-1/2 Polinom z 2 ničlami.

TG2-1/3 Polinom z več ničlami.

TG2-1/4 Polinom brez ničle.

TG2-1/5 Polinom, ki se enkrat dotakne abscise.

TG2-1/6 Polinom, ki se večkrat dotakne abscise.

Strategija Preverjanje robustnosti (Namen: Ali program pravilno procesira neveljavne vhodne podatke?)

TG3-1 Ali program pravilno procesira neveljavne vhodne podatke?

TG3-1/1 Ali program odpove pri $f(x)=x+x+x$?

TG3-1/2 Ali program odpove pri $f(x)=xxx$?

TG3-1/3 Ali program odpove pri $f(x)=1--1$?

TG3-1/4 Ali je menu robusten?

TG3-1/5 Ali je vnos polinoma robusten?

TG3-1/6 Ali je vnos parametrov za iskanje ničel robusten?

TG3-2 Ali je program odporen na neveljavno ime datoteke?

Strategija mejnih vrednosti (Ali program pravilno deluje pri mejnih vrednostih?)

TP4-1 Polinom z neskončnim številom ničel

TG4-2 Mejna vrednost eksponenta

Strategija strukturno testiranje

TP5-1 Klic funkcije (v izvorni kodi) DoubleSwap

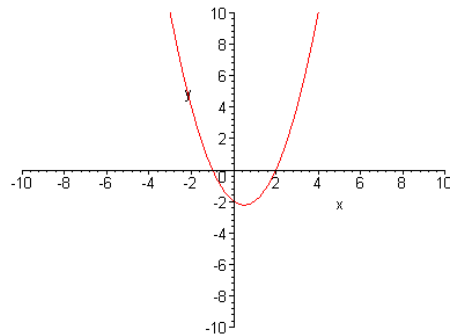
D4.7.3.1 Strategija 1: Preverjanje prisotnosti zahtev

4.7.3.1.1 TP1-1 Ali je prisotna najbolj kritična funkcionalnost? Ali je implementiran testni režim?- Demo primer iz specifikacij

Testni primer:	TP1- 1	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Polinomi
Garnitura:			Avtor:	Janez Ferin	

Začetno stanje	DOS (ali Windows Command Prompt), Na imeniku je datoteka tp1.txt, ki ima v prvi vrstici: $2x+1$	Strategija:	Preverjanje prisotnosti zahtev Tipični scenarij uporabe, pozitivni testni primer
Namen	Ali je prisotna najbolj kritična funkcionalnost? Ali je implementiran testni režim?	Opomba:	Polinom druge stopnje $y=x^2-x-2$. Testni primer je primeren za demonstracijo delovanja programa.
Konfiguracija sistema:	Windows XP, 256MB RAM, Duron 700 MHz...	Odpisnost od drugih testov:	Ni nobene odvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	vtipkamo POLINOMI –t		Izpisana pravilna verzija programa. Potem se javi glavni meni (zaslon 1). Ker je v testnem režimu, mora biti zaslon oštevilčen.	Stanje po zagonu programa	Glej Poročilo o nepravilnosti šte. 2/3
2	Izberem Vpis polinoma		Vnos polinoma (zaslon 2)		
3	Vnesem polinom	$3x^5 - 2x^3 + 2x^2 - 7$	Polinom prebere >> Glavni meni (zaslon 1)		
4	Izberem Izračun ničel in izris grafa funkcije		(zaslon 4) $f(x) = +3.000000x^5 - 2.000000x^3 + 2.000000x^2 - 7.000000$		
5	Izberem spodnjo mejo intervala iskanja (min x iskanja)	-10			
6	Izberem zgornjo mejo intervala iskanja (max x iskanja)	10			
7	Izberem natančnost iskanja ničel	0.001	Izpisane so tabelirane vrednosti polinoma v intervalu (-10,10) $f(-10.000000) = -297807.000000$; $f(-9.000000) = -175534.000000$; $f(-8.000000) = -97159.000000$; : $f(10.000000) = 298193.000000$;		

8	nadaljujem	[Enter]	Izpisane so: min x iskanja ničle polinoma $f(1.204042)=0$	Najdene morajo biti vse ničle in to z največjo napako v vrednosti po x osi, kot smo določili. V prototipu je privzeta maksimalna napaka 0.0001.	
9	nadaljujem	[Enter]	(zaslon 5) graf polinoma se izriše $f(x)=x^2-x-2$ (xmin=-10 xmax=10 ymin=-10 ymax=10  Prompt Enter	Graf je prikazan v ASCII načinu in je približek tega levo. V prototipu je vdelano fiksno območje -10..10 po obeh oseh.	
10	nadaljujem	0	Glavni meni (zaslon 1)		
11	Zahtevam branje polinoma iz datoteke	2	Meni (zaslon 3)		
12		tp1.txt	Glavni meni (zaslon 1). Pri točki 3 je izpisan polinom.		
13	Ponovim korake od 3 do 10		Izpis tabele, grafa in ničel		
14	Zahtevam izpis pomoči	*	Izpiše pomoč za glavni menu.		
15	meni – izberemo izhod	0	DOS prompt		

4.7.3.1.2 TP1-2 Ali je vedno prisotna vgrajena pomoč?

Testni primer:	TP1- 2	OPIS TESTNEGA PRIMERA	Verzija:	V 1.0	Projekt: Polinomi
Garnitura:			Avtor:	Janez Ferin	

Začetno stanje	Glej TP1-1	Strategija:	Preverjanje prisotnosti zahtev, pozitivni testni primer. V vsakem menuju zahtevaj izpis pomoči.
Namen	Ali je vedno prisotna vgrajena pomoč?	Opomba:	
Konfiguracija sistema:	Glej TP1-1	Odvisnost od drugih testov:	Ni nobene odvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
			Glavni meni (zaslon 1)	Stanje po zagonu programa	
1	Zahtevamo izpis pomoči	*	Program mora izpisati splošna navodila za uporabo osnovnih funkcij programa (za funkcije dostopne iz glavnega menija)		
2	Izberem Vpis polinoma in zahtevamo izpis pomoči	1	Vnos polinoma (zaslon 2)		
4	Vnesem nek polinom, izberemo izračun ničel in izris				
5	Ko je potrebno vnesti min x, zahtevamo pomoč	*	program izpiše pomoč za to vnosno polje		
7	min x iskanja	-10			
8	Ko je potrebno vnesti max x, zahtevamo pomoč	*	program izpiše pomoč za vnosno polje »max x iskanja«		
9	max x iskanja	10			
10	natančnost iskanja ničel	*	program izpiše pomoč za vnosno polje »natančnost iskanja ničel«		
11	Nadaljujem, dokler ni izrisan graf		program izpiše tabelo polinoma		
13	zahtevamo izpis pomoči	*	Program izpiše pomoč za meni za izris grafa		
14	zapustimo program		DOS prompt		

4.7.3.1.3 TP1-3 Ali izračunava ničle z zahtevano natančnostjo? Ali je dovolj hiter?

Testni primer:	TP1- 3	OPIS TESTNEGA PRIMERA	Verzija:	V 1.0	Projekt: Polinomi
Garnitura:			Avtor:	Janez Ferin	

Začetno stanje	Glej TP1-1	Strategija:	Preverjanje prisotnosti zahtev, pozitivni testni primer
Namen	Ali izračunava ničle z zahtevano natančnostjo? Ali je dovolj hiter?	Opomba:	
Konfiguracija sistema:	Glej TP1-1	Odvisnost od drugih testov:	Ni nobene odvisnosti

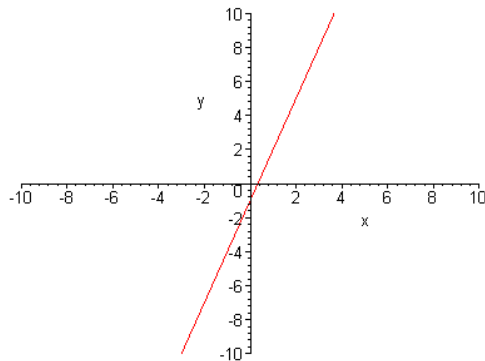
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	Opomba	opomba za testno poročilo
1	vnos polinoma	$3x-1$	program sprejme polinom $f(x)=3x-1$		
2	meni – izberemo izračun ničel in izris...	3			
3	min x iskanja	-10			
4	max x iskanja	10			
5	vnos zelene natančnosti in izris grafa	1	Program nadaljuje (z izpisom ničel). Ničla je pri $x=1/3$. Napaka pri izračunu ničle mora biti manjša od 1!	Opazujemo napako pri izračunu ničle!	
6	Ponovimo iskanje ničel z večjo natančnostjo	0.0001	Ničla je pri $x=1/3$. Napaka pri izračunu ničle mora biti manjša od 0.0001!	Opazujemo napako pri izračunu ničle!	
7	vnos zelene natančnosti	0.0000001	Ničla je pri $x=1/3$. Napaka pri izračunu ničle mora biti manjša od 0.0000001! Izračun se mora izvršiti prej kot v 5 sekundah.	Opazujemo napako pri izračunu ničle! Merimo čas.	
8	Končamo z uporabo		DOS prompt		

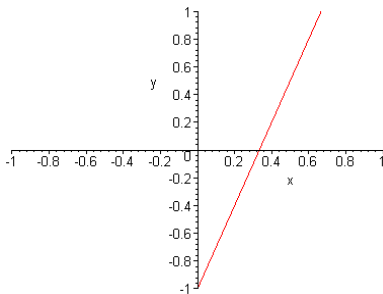
D4.7.3.2 Strategija 2: Ekvivalentni razredi polinomov.

4.7.3.2.1 TG2-1/1 Polinom z 1 ničlo.

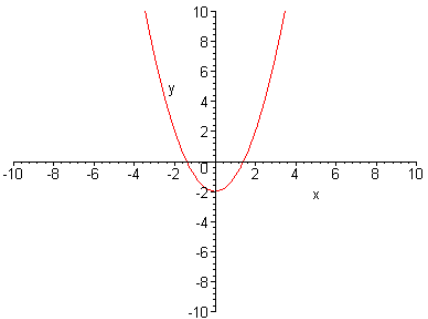
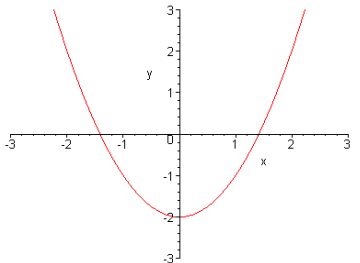
Garnitura:	TG2- 1	OPIS TESTNE GARNITURE	Verzija:	V 1.0	Projekt: Polinomi
Število testnih primerov:	6		Avtor:	Janez Ferin	

Začetno stanje	DOS (ali Windows Command Prompt), vtipkamo POLINOMI -t, izberemo vnos polinoma.	Strategija:	Ekvivalentni razredi polinomov, Negativni testni primeri
Namen	Ali izračuna ničle za vsak ekvivalentni razred? Ali upošteva spremembo območja risanja?	Opomba:	
Konfiguracija sistema:	Glej TP1-1	Odvisnost od drugih testov:	Ni nobene odvisnosti

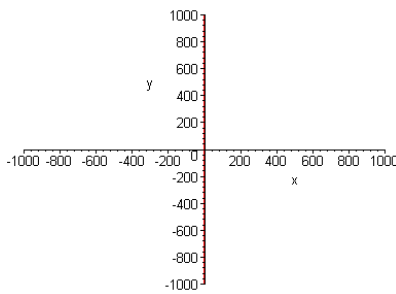
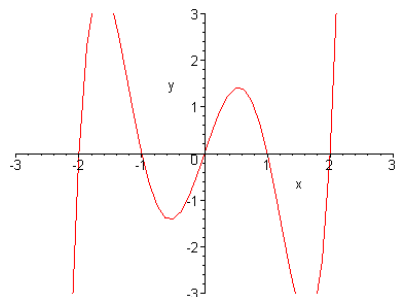
testni primer	TG2-1/1		Strategija /namen	Ekvivalentni razredi polinomov. Polinom z 1 ničlo.	
začetno stanje	glej opis testne garniture		odvisnost od prejšnjih testov	je ni	
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba	opomba za testno poročilo
1	vnos polinoma	$3x-1$	program sprejme polinom $f(x)=3x-1$		
2	vnos območja in izbrane natančnosti	min x =-10 max x = 10 natančnost=0.0000001	Izpisati mora tabelo za x v [-10..10], kjer $f(x)=3x-1$ Ničla polinoma je pri $x=1/3$, pričakujemo izpis na 6 natančnih decimalnih mest, torej 0.333333... 		
3	v meniju pri risanju grafa izberemo spremembo območja risanja	1	program omogoči spremembo mej risanja		

4		min x =-1, max x=1	<p>Program nadaljuje s ponovnim izrisom grafa. Graf mora ustrezati temu:</p> 		
5	v meniju za izris grafa izberemo vrnitev v glavni meni in nato konec	0	DOS prompt.		

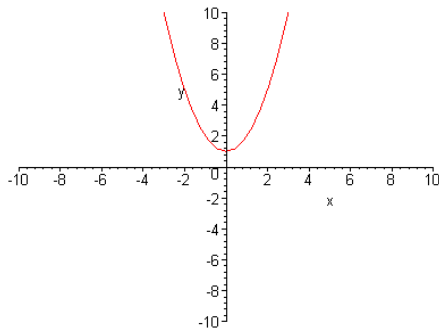
4.7.3.2.2 TG2-1/2 Polinom z 2 ničloma.

testni primer	TG2-1/2		strategija /namen	Ekvivalentni razredi polinomov. Polinom z 2 ničloma. Negativni test	
začetno stanje	glej opis testne garniture		odvisnost od prejšnjih testov	je ni	
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba	opomba za testno poročilo
1	vnos polinoma	x^2-2			
2	Vnesemo območje in natančnost	min x= -10 max x=10 natančnost=0.001	Izpisati mora tabelo za x v [-10..10] mora najti ničli $-\sqrt{2}, \sqrt{2}$ (-1.414213562, 1.414213562) vsaj na dve decimalni mesti natančno		
3	Izrišemo graf		graf mora ustrezati: 		
4	v meniju pri risanju grafa izberemo novo območje risanja	min x= -3 max x=3 natančnost=0.001	Graf mora ustrezati temu: 		
5	v meniju za izris grafa izberemo vrnitev v glavni meni in nato konec		DOS prompt.		

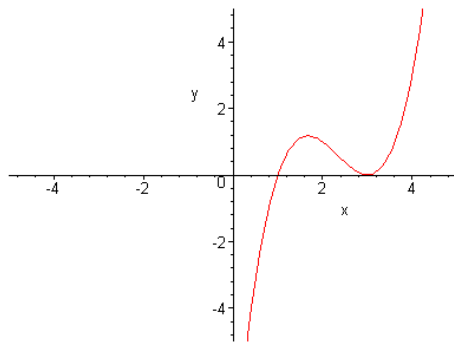
4.7.3.2.3 TG2-1/3 Polinom z več ničlami.

testni primer		TG2-1/3		strategija /namen	Ekvivalentni razredi polinomov. Polinom z več ničlami. Negativni test	
začetno stanje		glej opis testne garniture		odvisnost od prejšnjih testov	je ni	
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba	opomba za testno poročilo	
1	vnos polinoma	x^5-5x^3+4x				
2	Vnesemo območje in natančnost	min x= -1000 max x=1000 natančnost=0.0000001	Program izpiše tabelo polinoma. Izpisati mora tabelo za x v [-1000..1000]. Najti mora ničle -2, -1, 0, 1, 2 z izbrano natančnostjo (6 dec. mest)			
3	Izrišemo graf		graf mora ustrezati: 			
4	v meniju pri risanju grafa izberemo novo območje risanja	min x= -3 max x=3 natančnost=0.001	Graf mora ustrezati temu: 			
5	v meniju za izris grafa izberemo vrnitev v glavni meni in nato konec.	0	DOS prompt.			

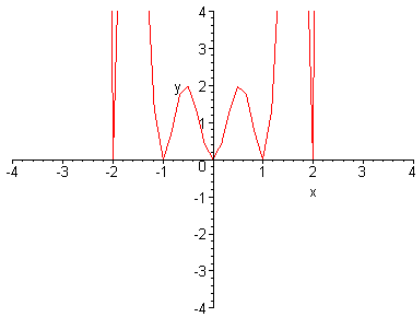
4.7.3.2.4 TG2-1/4 Polinom brez ničle.

testni primer		TG2- 1/4		strategija /namen		Ekvivalentni razredi polinomov: polinom brez ničle negativni testni primer. Negativni test.	
začetno stanje		glej opis testne garniture		odvisnost od prejšnjih testov		je ni	
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost		Opomba	opomba za testno poročilo	
1	vnos polinoma	x^2+1	program sprejme polinom $f(x)=x^2+1$				
2	Vnesemo območje in natančnost	min x= -10 max x=10 natančnost=0.001	Izpisati mora tabelo za x v [-10..10] in nobene ničle.				
3	izrišemo graf		program nadaljuje (z izpisom grafa) graf mora ustrezati: 				
4	v meniju za izris grafa izberemo vrnitev v glavni meni in nato konec.	0	DOS prompt.				

4.7.3.2.5 TG2-1/5 Polinom, ki se enkrat dotakne abscise.

testni primer		TG2-1/5		strategija		Ekvivalentni razredi polinomov. Polinom, ki se enkrat dotakne abscise. Negativni test
začetno stanje		glej opis testne garniture		odvisnost od prejšnjih testov		je ni
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba	opomba za testno poročilo	
1	vnos polinoma	$x^3 - 7x^2 + 15x - 9$	$f(x) = x^3 - 7x^2 + 15x - 9$			
2	Vnesemo območje in natančnost	min x= -1000 max x=1000 natančnost=0.000001	Izpisati mora tabelo za x v [-1000..1000] Najti mora ničle 1 in 3 (dvojna ničla) z izbrano natančnostjo (6 dec. mest)		Glej Poročilo o nepravilnosti števil. 1/4 Glej Poročilo o nepravilnosti števil. 2/1	
3	izrišemo graf		graf mora ustrezati: 			
4	v meniju za izris grafa izberemo vrnitev v glavni meni in nato konec.	0	DOS prompt.			

4.7.3.2.6 TG2-1/6 Polinom, ki se večkrat dotakne abscise.

testni primer		Strategija /namen		Ekvivalentni razredi polinomov. Polinom, ki se večkrat dotakne abscise. Negativni test	
začetno stanje		odvisnost od prejšnjih testov		je ni	
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba	opomba za testno poročilo
1	vnos polinoma	$x^{10}-10x^8+33x^6-40x^4+16x^2$	$f(x)=x^{10}-10x^8+33x^6-40x^4+16x^2$		
2	Vnesemo območje in natančnost	min x= -1000 max x=1000 natančnost=0.000001	Izpisati mora tabelo za x v [-1000..1000]. Mora najti ničle -2, -1, 0, 1 in 2. Vse ničle so dvojne.		Glej Poročilo o nepravilnosti šte. 1/5 Glej Poročilo o nepravilnosti šte. 2/2
3	izrišemo graf		program nadaljuje (z izpisom grafa) graf mora ustrezati: 		
4	v meniju za izris grafa izberemo vrnitev v glavni meni in nato konec.	0	DOS prompt.		

D4.7.3.3 Strategija 3: Preverjanje robustnosti (Namen: Ali program pravilno procesira neveljavne vhodne podatke?)

Garnitura:	TG3- 1	OPIS TESTNE GARNITURE	Verzija:	V 1.0	Projekt: Polinomi
Število testnih primerov:	6		Avtor:	Janez Ferin	

Začetno stanje	DOS (ali Windows Command Prompt), vtipkamo POLINOMI -t, izberemo vnos polinoma.	Strategija:	Ekvivalentni razredi polinomov, Negativni testni primeri
Namen	Ali je odporen na neveljavne vhodne podatke?	Opomba:	
Konfiguracija sistema:	Glej TP1-1	Ovisnost od drugih testov:	Ni nobene odvisnosti

4.7.3.3.1 TG3-1/1 Ali program odpove pri $f(x)=x+x+x$?

testni primer		TG3- 1/1		Strategija/ namen		Preverjanje robustnosti opisa polinoma, negativni testn primer	
začetno stanje		glej opis testne garniture		odvisnost od prejšnjih testov		je ni	
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost		opomba		opomba za testno poročilo
1	Vnos polinoma	x+x+x	program zavrne vnos ali sprejme polinom 3x				
2	meni – izberemo izhod	0	DOS prompt.				

4.7.3.3.2 TG3-1/2 Ali program odpove pri $f(x)=xxx$?

testni primer		TG3-1/2		Strategija/ namen		Preverjanje robustnosti opisa polinoma, negativni testni primer	
začetno stanje		glej opis testne garniture		odvisnost od prejšnjih testov		je ni	
korak	akcija / ime vhodne spremenljivke		vrednost	pričakovana reakcija/vrednost		opomba	opomba za testno poročilo
1	Vnos polinoma		xxx	program zavrne vnos ali sprejme polinom x^3			Glej Poročilo o nepravilnosti števil. 1/1
2	meni – izberemo izhod		0	DOS prompt.			

4.7.3.3.3 TG3-1/3 Ali program odpove pri $f(x)=1--1$?

testni primer		TG3-1/3		Strategija/ namen		Preverjanje robustnosti opisa polinoma, negativni testni primer	
začetno stanje		glej opis testne garniture		odvisnost od prejšnjih testov		je ni	
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost		opomba		opomba za testno poročilo
1	Vnos polinoma	1--1	program zavrne vnos ali sprejme polinom f(x)=2				Glej Poročilo o nepravilnosti števil. 1/2
2	meni – izberemo izhod	0	DOS prompt.				

4.7.3.3.4 TG3-1/4 Ali je menu robusten?

testni primer		TG3-1/4		Opomba / namen		Ali je menu robusten, če izbiramo neveljavne odgovore?
začetno stanje		glej opis testne garniture		odvisnost od prejšnjih testov		je ni
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba	opomba za testno poročilo	
1	Meni	3,4 ,strela	program zavrne vnos	Namesto vejice stisni [enter]		
2	Meni	*	program izpiše pomoč			
3	meni – izberemo izhod	0	program zaključi			

4.7.3.3.5 TG3-1/5 Ali je vnos polinoma robusten- $f(x)$ ima neveljaven en člen?

testni primer		TG3-1/5		Opomba / namen		Preverjanje robustnosti opisa polinoma, negativni testni primer.
začetno stanje		glej opis testne garniture		Odvisnost od prejšnjih testov		je ni
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba	opomba za testno poročilo	
1	zahtevamo vnos polinoma in nato vtipkamo neveljavne opise:	1	Zaslon 2			
2		strela	program zavrne vnos			
4		y^3	program zavrne vnos			
6		$\sin(x)$	program zavrne vnos			
7		1E40	program zavrne vnos			
8		-1E40	program zavrne vnos			
9		$2x^{300000000000}$	program zavrne vnos			
10	meni – izberemo izhod	0	program zaključi			

4.7.3.3.6 TG3-1/6 Ali je vnos parametrov za iskanje ničel robusten?

testni primer		TG3-1/6		Opomba / namen		Ali je vnos parametrov za iskanje ničel robusten?
začetno stanje		glej opis testne garniture		odvisnost od prejšnjih testov		je ni
korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba	opomba za testno poročilo	
1	Vnesemo polinom $f(x)=x$ in izberemo izračun ničel...					
3	min x iskanja je nedovoljena vrednost	strela	program mora zavrniti vnos	Neveljavna vrednost	Glej Poročilo o nepravilnosti števil. 1/3	
4	iskanja	min $x = 1000$ max $x = -1000$	Vtipkam veljavno vrednost, da lahko nadaljujem			
6	vnos natančnosti	kaj	program zavrne vnos	Neveljavna vrednost		
7	vnos natančnosti	0.001	Vtipkam veljavno vrednost, da lahko nadaljujem			

8	Sprememba območja (nedovoljena vrednost)	mir	program zavrne vnos		
9	Končam test		program nadaljuje (z izpisom grafa)		

4.7.3.3.7 TG3-2 Ali je program odporen na neveljavno ime datoteke?

Garnitura:	TG 3-2	OPIS TESTNE GARNITURE	Verzija:	V 1.0	Projekt: Polinomi
Število testnih primerov:	3		Avtor:	Janez Ferin	

Začetno stanje	Datoteka x:\poli.not ne sme obstajati. DOS (ali Windows Command Prompt), vtipkamo POLINOMI –t in zahtevamo branje polinoma iz datoteke.	Strategija:	Preverjanje robustnosti, negativni test
Namen	Nepravilna imena, oziroma datoteka ne obstaja.	Opomba:	
Konfiguracija sistema:	Windows XP, 256MB RAM, Duron 700 MHz...	Odvisnost od drugih testov:	Ni nobene odvisnosti

testni primer	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	Opomba / namen	opomba za testno poročilo
TG3-2/1	Ime datoteke	x:\poli.not	program zavrne vnos	Niti disk niti datoteka ne obstajata)	
TG3-2/2	Ime datoteke	/dev/mnt/c/nogo	program zavrne vnos	ime datoteke ni zapisano v DOSovskem formatu	
TG3-2/3	ime datoteke	polinomi.exe	program zavrne vnos	datoteka ne vsebuje polinoma	
	meni – izberemo izhod	0	program zaključi		

D4.7.3.4 Strategija 4: mejne vrednosti

4.7.3.4.1 TP4-1 Polinom z neskončnim številom ničel

Testni primer:	TP4-1	OPIS TESTNEGA PRIMERA	Verzija:	V 1.0	Projekt: Polinomi
Garnitura:			Avtor:	Janez Ferin	

Začetno stanje	Glej testni primer TP1-1	Strategija:	Mejne vrednosti, negativni testni primer.
Namen	Ali pravilno deluje, če imamo neskončno ničel.	Opomba:	
Konfiguracija sistema:	Glej testni primer TP1-1	Odvisnost od drugih testov:	Ni nobene odvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	Opomba	opomba za testno poročilo
1	vnos polinoma	0	program sprejme polinom $f(x)=0$		
2	Vnesemo območje in natančnost	min x= -3 max x=3 natančnost=0.001	Izpisati mora tabelo za x v [-10..10] (celoten graf leži na X osi)		Glej Poročilo o nepravilnosti števil. 1/6
3	v meniju za izris grafa izberemo vrnitev v glavni meni in nato konec.	0	DOS prompt.		

4.7.3.4.2 TG4-2 meja – vrednost eksponenta

Garnitura:	TG 4-2	OPIS TESTNE GARNITURE	Verzija:	V 1.0	Projekt: Polinomi
Število testnih primerov:	3		Avtor:	Janez Ferin	

Začetno stanje	Glej testni primer TP1-1	Strategija:	Mejne vrednosti, negativni testni primer.
Namen	Ali pravilno deluje, če uporabimo maksimalni integer.	Opomba:	
Konfiguracija sistema:	Glej testni primer TP1-1	Odvisnost od drugih testov:	Ni nobene odvisnosti

testni primer	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	Opomba / namen	opomba za testno poročilo
TG4-2/1	Vnesem polinom in zahtevam izračun ničle max. stopnja	$x^2, 147,483,647$	Rezultat x=0	2,147,483,64 je maksimalni integer	
TG4-2/2	max. stopnja+1	$x^2, 147,483,648$	Rezultat x=0 ali obvestilo sistema		
TG4-2/3	max. stopnja-1	$x^2, 147,483,646$	Rezultat x=0 ali obvestilo sistema		

D4.7.3.5 Strategija 5: Strukturno testiranje

4.7.3.5.1 TP5-1 Klic funkcije (v izvorni kodi) DoubleSwap.

Testni primer:	TP5-1	OPIS TESTNEGA PRIMERA	Verzija:	V 1.0	Projekt: Polinomi
Garnitura:			Avtor:	Janez Ferin	

Začetno stanje	Instrumentiran program testiramo z vsemi testnimi primeri. Funkcija DoubleSwap ni bila nikoli klicana. DOS (ali Windows Command Prompt), vtipkamo POLINOMI –t.	Strategija:	strukturno testiranje- Predpostavka o napaki: v stavkih, ki so še neizvršeni, se nahaja napaka.
Namen	Klicati funkcijo (v izvorni kodi) DoubleSwap.	Opomba:	Pokrivanje nepokritih stavkov oziroma rutin. Funkcija(v izvorni kodi) DoubleSwap ni bila nikoli klicana.
Konfiguracija sistema:	Glej TP1-1	Odvisnost od drugih testov:	Z instrumentirano verzijo programa je potrebno izvesti vse teste, ki ne spadajo v to strategijo.

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	vnos polinoma	x	f(x)=x		
2	vnos območja in izbrane natančnosti	min x =10 max x =-10 natančnost=0.001		Z analizo izvorne kode je bilo ugotovljeno, da bo program v primeru, da je spodnja meja večja od zgornje, ti dve meji zamenjal s pomočjo funkcije DoubleSwap. (Ta funkcija zamenja vrednost dveh spremenljivk.)	
3	Izrišemo graf in zapustimo program	10	Preverimo pokritost rutin: Rutina DoubleSwap mora biti najmanj enkrat uporabljena.		

D5 POROČILO O PREVERJANJU

D5.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument Poročilo o preverjanju	
verzija	1
datum	22.4.2002
lokacija	datoteka Projekt_Polinomi.doc v podmapi »dokumentacija« mape projekta
Odgovorna oseba	
	Janez Ferin
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve
	Plan projekta, V0.9
	Sistemske specifikacije V 1.0
	Uporabniški priročnik V 1.0
	Testni primeri 1.0

D5.2 POVZETEK

Produkt smo preverili v skladu s standardom CVVS 2-2000 in načrtom zagotavljanja kakovosti opisanem v planu projekta (točka D2.3). Uporabili smo testne primere podrobneje opisane v dokumentu Testni primeri.

D5.3 IZVEDENE AKTIVNOSTI IN UGOTOVITVE

Izvedene so bile vse aktivnosti, ki so bile predvidene v Planu projekta:

- splošni pregled dokumentov oziroma vmesnih produktov
- evalvacija prototipa
- pregled izvirne kode
- testiranje končnega produkta

D5.3.1 Splošni pregled dokumentov oziroma produktov, razen izvršljivega programa in izvirne kode

Pregledani so bili vsi dokumenti. Termini pregledov so razvidni iz testne matrike. Vse najdene neustreznosti so bile odpravljene. Vsi dokumenti **ustrezajo** standardu.

3.2. Evalvacija prototipa

Prototip smo preverili s pregledom in s testiranjem. Preverjeno je bilo, ali je programer kodiral v skladu z zahtevanim stilom. Vse nepravilnosti glede stila kodiranja so bile ustno sporočene programerju. Za evalvacijo smo uporabili modificirani testni primer TP1 – zaradi tega, ker vse funkcije v prototipu še niso implementirane. Ugotovljene nepravilnosti so bile ustno sporočene razvijalcu programske opreme.

3.3. Pregled izvirne kode (verzija 2.0)

Stopnja dokumentiranosti znaša: 3.06 besed na vrstico /programske kode. Za meritev smo uporabili orodje Winmetr V2.0. Izpis je v dodatku. Logična kompleksnost (LC1) posamezne rutine je bila vedno manjša od 20. Izvorna koda je bila pregledana in je v skladu s standardom cV&Vs.

3.4. Testiranje programa

Testiranje je potekalo na računalniku PC s sistemom Windows XP. Uporabili smo orodje InstrWin-99. Testirali smo dve verziji programa Polinomi.exe (V1.0 in V2.0).

1. Seznam uporabljenih strategij:

Pri načrtovanju testnih primerov so bile uporabljene naslednje strategije:

- a) prisotnost zahtev (Z)
- b) preverjanje robustnosti (R)
- c) ekvivalentni razredi polinomov (E)
- d) mejne vrednosti (M)
- e) strukturno testiranje (S)

3. Vložen napor:

Glede na izbrane strategije in terminalno kriterijsko funkcijo, je bilo uporabljenih **18 testnih primerov**. Testni primeri so opisani v dokumentu Testni primeri.

4. Dosežena temeljitost:

Planirana terminalna kriterijska funkcija: (pg1 IN pg2) ALI pg3.

pg1. 100% pokritost zahtev

pg2. Vsaka funkcija v izvorni kodi mora biti klicana najmanj enkrat.

pg3. Ko preteče predvideno obdobje, ki je namenjeno testiranju.

S testiranjem smo končali, ko je bilo zadoščeno pogojema pg1 in pg3.

Dosežena pokritost kode¹:

100% pokritost zahtev (13 zahtev)

97.14% oziroma klicanih je bilo 34 od 35 funkcij

Funkcija IntSwap ni bila nikoli klicana. Zaradi splošne zasnove modula risar, ta funkcija, ki skrbi za pravilno orientacijo vhodnih podatkov, ni bila nikoli izvedena. Dejansko bi jo lahko, kar se tiče projekta polinomi, tudi odstranili. Funkcijo bi lahko dosegli le z izoliranim testiranjem modula risar.

5. Ugotovitve:

V prvi verziji je bilo odkritih 6 nepravilnosti. Od tega smo štiri takoj odpravili. Po opravljenem testiranju in popravljanju smo dvema nepravilnostima dodelili status hibe. Pravzaprav gre za eno samo nepravilnost, ki smo jo zaznali z dvema testnima primeroma. Ta hiba se nanaša na kritični del funkcionalnosti – iskanje ničel. Program namreč ne najde vseh ničel. Predvsem so problem dvojne ničle (in druge ničle sode stopnje), v katerih se graf polinoma le dotakne osi X. Gre za omejitve numeričnih metod iskanja ničel polinomov. Edini primer, ko so te metode 100% zanesljive, je, kadar ima vrednost funkcije na spodnji in zgornji meji intervala iskanja ničel različen predznak. Takrat tudi program polinomi ničle zagotovo najde s poljubno natančnostjo. Ker ne obstaja niti analitični način za popolnoma uspešno iskanje ničel polinomov višje stopnje, lahko odpoved kljub kritičnosti pripišemo k hibam. Vse druge resne nepravilnosti so bile odpravljene.

Zadnja verzija:

V zadnji verziji (oznaka V2.0) smo odkrili 3 nepravilnosti, ki so **dobili status hibe** (glej razlago zgoraj ter Poročilo o najdenih neustreznostih, Pregledni seznam vseh opaženih neustreznosti).

Uspešnost odpravljanja je bila preverjena z regresijskimi testi. V obeh verzijah skupaj je bilo odkritih 9 nepravilnosti.

Izpis prevajalnika ne vsebuje nobenih napak in opozoril.

¹ Uporabi, če je na voljo ustrezna programska podpora.

D5.4 POROČILO O NAJDENIH NEPRAVILNOSTIH

Pregledni seznam vseh opaženih neustreznosti:

Nepravilnost štev.	Testni primer	Opis nepravilnosti	Resnost	Nepravilnost je bila odkrita v verziji	Končni status nepravilnosti	Opomba
1/1	TG3-1/2	$f(x)=xxx$ napačno sprejme kot $f(x)=x^3$	2	V1.0	odpravljeno v verziji 2.0	
1/2	TG3-1/3	$f(x)=1--1$ napačno sprejme kot $f(x)=-1$	2	V1.0	odpravljeno v verziji 2.0	
1/3	TG3-1/6	Program pri vnosu intervala tabeliranja omogoča vnos nesmiselnih vrednosti	1	V1.0	odpravljeno v verziji 2.0	
1/4	TG2-1/5	Program ne najde dvojnih ničel	3	V1.0	hiba	(minimalno izboljšano v verziji 2.0)
1/5	TG2-1/6	Program ne najde dvojnih ničel	3	V1.0	hiba	(minimalno izboljšano v verziji 2.0)
1/6	TP4-1	Program ne najde ničel polinoma $f(x)=0$	3	V1.0	odpravljeno v verziji 2.0	(bistveno izboljšano v verziji 2.0, kjer program izpiše prvih 500 ničel in opozorilo)
2/1	TG2-1/5	Program ne najde dvojnih ničel	3	V2.0	hiba	Prenešena nepravilnost 1/4 iz verzije V1.0
2/2	TG2-1/6	Program ne najde dvojnih ničel	3	V2.0	hiba	Prenešena nepravilnost 1/5 iz verzije V1.0
2/3	TP4-1	Izpisana je napačna verzija programa	1	V2.0	hiba	

Odkrili smo skupaj 9 različnih nepravilnosti na 1000 vrstic izvirne kode.

D5.5 PRILOGE

D5.5.1 Poročila o neustreznostih



POROČILO O NEPRAVILNOSTI

Forma: ONEU

V3.0s

1) PROJEKT: POLINOMI 2) NEPRAVILNOST ŠTEV.: 1/1
3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0
5) KONFIGURACIJA: SYSTEM1.TXT

6) VRSTA NEPRAVILNOSTI (1-3): 7) RESNOST (1-3):
☒ 1 Nepravilno implementirano 1 - Nepomembna
☐ 2 Ni implementirano ☒ 2 Resna
☐ 3 Ni bilo zahtevano 3 - Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: ZASLON ŠT. 2 - VNOS POLINOMA

9) KRATEK OPIS NEPRAVILNOSTI:

Polinom xxx interpretira kot 3x. Pravilno
je: x³

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)?

11) POGOSTOST POJAVLJANJA: 1 zelo pogosto, ☒ 2 občasno, 3 redko

12) TESTNI PRIMER/POSTOPEK:

GLEJ TG 3-1/2

13) PRILOGE:

14) POROČEVALEC: J. FERIN

15) DATUM: 25.3.2002

16) ODGOVORNA OSEBA J. FERIN Izpolni vodja

17) PRIORITETA: ☒ 1 Takoj odpraviti 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 1
(glej polje 20)

19) Podpis vodje Ferin

Izpolni razvojna skupina

20) NEPRAVILNOST JE (1-9):

<input checked="" type="radio"/> 1 Odpravljena	4 Zavrnjena	7 Sprejeta
2 Prestavljena (hiba)	5 Preklicana	8 Duplikat
3 Neponovljiva	6 Ignorirana	9 Potrebne so dodatne inf.

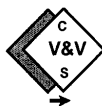
21) OPOMBA

22) NEPRAVILNOST ODPRASIL M. ŠTUKOVNIK

23) DATUM 26.3.2002

24) POPRAVILO PREVERIL J. FERIN

25) DATUM 29.3.2002



POROČILO O NEPRAVILNOSTI

Forma: ONEU

V3.0s

1) PROJEKT: POLINOMI 2) NEPRAVILNOST ŠTEV.: 1/2

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0

5) KONFIGURACIJA: SISTEM1.TXT

6) VRSTA NEPRAVILNOSTI (1-3):

- ☒ 1 Nepravilno implementirano
☐ 2 Ni implementirano
☐ 3 Ni bilo zahtevano

7) RESNOST (1-3):

- ☐ 1 - Nepomembna
☒ 2 Resna
☐ 3 - Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: ZASLON ŠT. 2 - VNOS POLINOMA

9) KRATEK OPIS NEPRAVILNOSTI: _____

Polinom $f(x) = 1 - -1$ napačno sprejme kot
 $f(x) = -1.00$. Vnos bi moral zavrniti.

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)?

11) POGOSTOST POJAVLJANJA: 1 zelo pogosto, ☒ 2 občasno, 3 redko

12) TESTNI PRIMER/POSTOPEK: _____

GLEJ TG3-1/3

13) PRILOGE: /

14) POROČEVALEC: J. FERIN

15) DATUM: 25.3.2002

16) ODGOVORNA OSEBA J. FERIN Izpolni vodja

17) PRIORITETA ☒ 1 Takoj odpraviti 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 1
(glej polje 20)

19) Podpis vodje Ferin

Izpolni razvojna skupina

20) NEPRAVILNOST JE (1-9):

- | | | |
|--|--------------|----------------------------|
| <input checked="" type="radio"/> 1 Odpravljena | 4 Zavrnjena | 7 Sprejeta |
| 2 Prestavljena (hiba) | 5 Preklicana | 8 Duplikat |
| 3 Neponovljiva | 6 Ignorirana | 9 Potrebne so dodatne inf. |

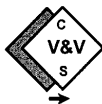
21) OPOMBA /

22) NEPRAVILNOST ODPRAVIL M. ŠTUKOVNIK

23) DATUM 26.3.2002

24) POPRAVILO PREVERIL J. FERIN

25) DATUM 29.3.2002



POROČILO O NEPRAVILNOSTI

Forma: ONEU

V3.0s

1) PROJEKT: POLINOMI 2) NEPRAVILNOST ŠTEV.: 1/3

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0

5) KONFIGURACIJA: _____

6) VRSTA NEPRAVILNOSTI (1-3):

- ☒ 1) Nepravilno implementirano
☐ 2) Ni implementirano
☐ 3) Ni bilo zahtevano

7) RESNOST (1-3):

- ☐ 1 - Nepomembna
☐ 2 - Resna
☐ 3 - Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: IZRAČUNI NIČEL - ZASLON ŠT. 4.

9) KRATEK OPIS NEPRAVILNOSTI:

PRI VNOSU INTERVALA TABELIRANJA
JE MOŽEN VNOS NESMISELNIH VREDNOSTI

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)?

11) POGOSTOST POJAVLJANJA: 1 zelo pogosto, 2 občasno, ☒ 3 redko

12) TESTNI PRIMER/POSTOPEK :

GLEJ T63-1/6

13) PRILOGE: /

14) POROČEVALEC: J. FERIN

15) DATUM: 25.3.2002

16) ODGOVORNA OSEBA J. FERIN Izpolni vodja

17) PRIORITETA: 1 Takoj odpraviti ☒ 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 1
(glej polje 20)

19) Podpis vodje Ferin

Izpolni razvojna skupina

20) NEPRAVILNOST JE (1-9):

- | | | |
|---|-------------------------------------|---|
| <input checked="" type="radio"/> 1) Odpravljena | <input type="radio"/> 4) Zavrnjena | <input type="radio"/> 7) Sprejeta |
| <input type="radio"/> 2) Prestavljena (hiba) | <input type="radio"/> 5) Preklicana | <input type="radio"/> 8) Duplikat |
| <input type="radio"/> 3) Neponovljiva | <input type="radio"/> 6) Ignorirana | <input type="radio"/> 9) Potrebne so dodatne inf. |

21) OPOMBA /

22) NEPRAVILNOST ODPRAVIL M. ŠTUKOVNIK

23) DATUM 26.3.2002

24) POPRAVILO PREVERIL J. FERIN

25) DATUM 29.3.2002



POROČILO O NEPRAVILNOSTI

Forma: ONEU

V3.0s

1) PROJEKT: POLINOMI 2) NEPRAVILNOST ŠTEV.: 1/4

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0

5) KONFIGURACIJA: SISTEM1.TXT

6) VRSTA NEPRAVILNOSTI (1-3):

- ☒ 1 Nepravilno implementirano
☐ 2 Ni implementirano
☐ 3 Ni bilo zahtevano

7) RESNOST (1-3):

- ☐ 1 - Nepomembna
☐ 2 - Resna
☒ 3 Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: IZPIS (IZRAČUN) NIČEL (ZASLON 4)

9) KRATEK OPIS NEPRAVILNOSTI:

PROGRAM NE NAJDE VSEH NIČEL NA PODANEM INTERVALU
Npr. $f(x) = x^3 - 7x^2 + 15x - 9$ NE NAJDE NIČLE $x=3$
V TEJ TOČKI SE POLINOM SAMO DOTAKNE
 $x=0.51$

10) JE PONOVLJIVOST ZAGOTOVLJENA ☒ (da) ne)?

11) POGOSTOST POJAVLJANJA: 1 zelo pogosto, 2 občasno, ☒ 3 redko

12) TESTNI PRIMER/POSTOPEK:

GLEJ TG 1/5

13) PRILOGE: /

14) POROČEVALEC: J. FERIN

15) DATUM: 25.3.2002

Izpolni vodja

16) ODGOVORNA OSEBA: J. FERIN

17) PRIORITETA: 1 Takoj odpraviti ☒ 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 2
(glej polje 20)

19) Podpis vodje

Ferin

Izpolni razvojna skupina

20) NEPRAVILNOST JE (1-9):

- | | | |
|--|------------------------------------|--|
| <input type="radio"/> 1 Odpravljena | <input type="radio"/> 4 Zavrnjena | <input type="radio"/> 7 Sprejeta |
| <input checked="" type="radio"/> 2 Prestavljena (hiba) | <input type="radio"/> 5 Preklicana | <input type="radio"/> 8 Duplikat |
| <input type="radio"/> 3 Neponovljiva | <input type="radio"/> 6 Ignorirana | <input type="radio"/> 9 Potrebne so dodatne inf. |

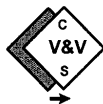
21) OPOMBA KER Z NUMERIČNO METODO NI MOGOČE ZANESLJIVO
NAJTI VSEH NIČEL, PREDLAGAMO STATUS HIBA

22) NEPRAVILNOST ODPRAVIL

23) DATUM

24) POPRAVILO PREVERIL J. FERIN

25) DATUM 29.3.2002



POROČILO O NEPRAVILNOSTI

Forma: ONEU

V3.0s

1) PROJEKT: POLINOMI 2) NEPRAVILNOST ŠTEV.: 1/5
3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0
5) KONFIGURACIJA: SISTEM1.TXT

6) VRSTA NEPRAVILNOSTI (1-3):

- ☒ 1 Nepravilno implementirano
☐ 2 Ni implementirano
☐ 3 Ni bilo zahtevano

7) RESNOST (1-3):

- ☐ 1 - Nepomembna
☐ 2 - Resna
☒ 3 Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: IZPIS (IZRAČUN) NIČEL (ZASLON 4)

9) KRATEK OPIS NEPRAVILNOSTI:

PROGRAM NE NAJDE DVOJNIT NIČEL

POLINOMA $f(x) = x^{10} - 10x^8 + 33x^6 - 40x^4 + 16x^2$

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)?

11) POGOSTOST POJAVLJANJA: 1 zelo pogosto, 2 občasno, 3 redko

12) TESTNI PRIMER/POSTOPEK:

GLEJ TG 2-1/6

13) PRILOGE:

14) POROČEVALEC: J. FERIN

15) DATUM: 25.3.2002

Izpolni vodja

16) ODGOVORNA OSEBA: J. FERIN

17) PRIORITETA: 1 Takoj odpraviti, 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI: 2

(glej polje 20)

19) Podpis vodje

Izpolni razvojna skupina

20) NEPRAVILNOST JE (1-9):

- | | | |
|--|--------------|----------------------------|
| 1 Odpravljena | 4 Zavrnjena | 7 Sprejeta |
| <input checked="" type="radio"/> 2 Prestavljena (hiba) | 5 Preklicana | 8 Duplikat |
| 3 Neponovljiva | 6 Ignorirana | 9 Potrebne so dodatne inf. |

21) OPOMBA: PREDLAGAMO STATUS HIBE

22) NEPRAVILNOST ODPRAVIL

23) DATUM

24) POPRAVILO PREVERIL: J. FERIN

25) DATUM: 29.3.2002



V3.0s

POROČILO O NEPRAVILNOSTI

Forma: ONEU

1) PROJEKT: POLINOMI 2) NEPRAVILNOST ŠTEV.: 1/6

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0

5) KONFIGURACIJA: SYSTEM1.TXT

6) VRSTA NEPRAVILNOSTI (1-3): 7) RESNOST (1-3):
☒ 1 Nepravilno implementirano 1 - Nepomembna
2 Ni implementirano 2 - Resna
3 Ni bilo zahtevano ☒ 3 Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: IZPIS (IZRAČUN) NIČEL - ZASLON 4

9) KRATEK OPIS NEPRAVILNOSTI: PROGRAM NE OPOZORI, DA
GRE ZA NESKONČNO ŠTEVILO NIČEL

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)?
11) POGOSTOST POJAVLJANJA: Zelo pogosto, 2 občasno, 3 redko

12) TESTNI PRIMER/POSTOPEK: glej TP 4-1

13) PRILOGE: /

14) POROČEVALEC: J. FERIN 15) DATUM: 25.2.2002

16) ODGOVORNA OSEBA: J. FERIN Izpolni vodja

17) PRIORITETA: Takoj odpraviti 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 1 19) Podpis vodje Ferin
(glej polje 20)

Izpolni razvojna skupina

20) NEPRAVILNOST JE (1-9):
☒ 1 Odpravljena 4 Zavrnjena 7 Sprejeta
2 Prestavljena (hiba) 5 Preklicana 8 Duplikat
3 Neponovljiva 6 Ignorirana 9 Potrebne so dodatne inf.

21) OPOMBA /

22) NEPRAVILNOST ODPRASIL M. ŠTUKOVNIK 23) DATUM 27.3.2002

24) POPRAVILO PREVERIL J. FERIN 25) DATUM 28.3.2002



POROČILO O NEPRAVILNOSTI

Forma: ONEU

V3.0s

1) PROJEKT: POLINOM1 2) NEPRAVILNOST ŠTEV.: 2/1

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 2.0

5) KONFIGURACIJA: SYSTEM1.TXT

6) VRSTA NEPRAVILNOSTI (1-3): 7) RESNOST (1-3):
☒ 1 Nepravilno implementirano 1 - Nepomembna
☐ 2 Ni implementirano 2 - Resna
☐ 3 Ni bilo zahtevano ☒ 3 Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: IZRAČUN NIČEL (ZASLON 5T.4)

9) KRATEK OPIS NEPRAVILNOSTI: PROGRAM NE NAJDE VSEH NIČEL

10) JE PONOVLJIVOST ZAGOTOVLJENA (dane)?
11) POGOSTOST POJAVLJANJA: Pozelo pogosto, 2 občasno, 3 redko
12) TESTNI PRIMER/POSTOPEK: glej T 92-116

13) PRILOGE: /

14) POROČEVALEC: J. FERIN 15) DATUM: 28.3.2002

16) ODGOVORNA OSEBA: J. FERIN Izpolni vodja

17) PRIORITETA: 1 Takoj odpraviti 2 Čim bo mogoče

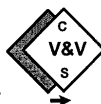
18) KONČNI STATUS NEPRAVILNOSTI 2 19) Podpis vodje Ferin
(glej polje 20) Izpolni razvojna skupina

20) NEPRAVILNOST JE (1-9):
1 Odpravljena 4 Zavrnjena 7 Sprejeta
☒ 2 Prestavljena (hiba) 5 Preklicana 8 Duplikat
3 Neponovljiva 6 Ignorirana 9 Potrebne so dodatne inf.

21) OPOMBA _____

22) NEPRAVILNOST ODPRAVIL _____ 23) DATUM _____

24) POPRAVILO PREVERIL _____ 25) DATUM _____



POROČILO O NEPRAVILNOSTI

Forma: ONEU

V3.0s

1) PROJEKT: POLINOMI 2) NEPRAVILNOST ŠTEV.: 2/2

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 2.0

5) KONFIGURACIJA: SISTEMA, TXT

6) VRSTA NEPRAVILNOSTI (1-3):

- ☒ 1 Nepravilno implementirano
☐ 2 Ni implementirano
☐ 3 Ni bilo zahtevano

7) RESNOST (1-3):

- ☐ 1 - Nepomembna
☐ 2 - Resna
☒ 3 Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: IZRAČUN NIČEL (ZASLON 4)

9) KRATEK OPIS NEPRAVILNOSTI: _____

OD PETIH NIČEL, PROGRAM
NAJDE LE ENO

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)?

11) POGOSTOST POJAVLJANJA: ☒ Zelo pogosto, 2 občasno, 3 redko

12) TESTNI PRIMER/POSTOPEK: _____

glej TG 4/7

13) PRILOGE: /

14) POROČEVALEC: J. FERIN 15) DATUM: 28.3.2002

16) ODGOVORNA OSEBA J. FERIN Izpolni vodja

17) PRIORITETA: 1 Takoj odpraviti 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 2 19) Podpis vodje Ferin
(glej polje 20)

Izpolni razvojna skupina

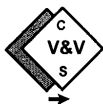
20) NEPRAVILNOST JE (1-9):

- | | | |
|--|------------------------------------|--|
| <input type="radio"/> 1 Odpravljena | <input type="radio"/> 4 Zavrnjena | <input type="radio"/> 7 Sprejeta |
| <input checked="" type="radio"/> 2 Prestavljena (hiba) | <input type="radio"/> 5 Preklicana | <input type="radio"/> 8 Duplikat |
| <input type="radio"/> 3 Neponovljiva | <input type="radio"/> 6 Ignorirana | <input type="radio"/> 9 Potrebne so dodatne inf. |

21) OPOMBA _____

22) NEPRAVILNOST ODPRAVIL _____ 23) DATUM _____

24) POPRAVILO PREVERIL _____ 25) DATUM _____



POROČILO O NEPRAVILNOSTI

Forma: ONEU

V3.0s

1) PROJEKT: POLINOMI 2) NEPRAVILNOST ŠTEV.: 2/3

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 2.0

5) KONFIGURACIJA: SYSTEM1.TXT

6) VRSTA NEPRAVILNOSTI (1-3):
☒ 1) Nepravilno implementirano
☐ 2) Ni implementirano
☐ 3) Ni bilo zahtevano

7) RESNOST (1-3):
☒ 1 - Nepomembna
☐ 2 - Resna
☐ 3 - Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: GLAVNI MENU (ZASLON ST. 1)

9) KRATEK OPIS NEPRAVILNOSTI: VERZIJA PROGRAMA NI
PRAVILNA. IZPIŠE V1.0 NAMESTO V2.0

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)? da

11) POGOSTOST POJAVLJANJA: Zelo pogosto, 2 občasno, 3 redko

12) TESTNI PRIMER/POSTOPEK: TP1

13) PRILOGE: /

14) POROČEVALEC: J. FERIN 15) DATUM: 28.3.2002

16) ODGOVORNA OSEBA: J. FERIN Izpolni vodja

17) PRIORITETA: 1 Takoj odpraviti 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 2 19) Podpis vodje _____
(glej polje 20)

Izpolni razvojna skupina

20) NEPRAVILNOST JE (1-9):
☒ 1 Odpravljena ☐ 4 Zavrnjena ☐ 7 Sprejeta
☐ 2 Prestavljena (hiba) ☐ 5 Preklicana ☐ 8 Duplikat
☐ 3 Neponovljiva ☐ 6 Ignorirana ☐ 9 Potrebne so dodatne inf.

21) OPOMBA _____

22) NEPRAVILNOST ODPRAVIL _____ 23) DATUM _____

24) POPRAVILO PREVERIL _____ 25) DATUM _____

D5.5.2 Podatki o kompleksnosti in komentiranosti

Kompleksnost in komentiranost smo izmerili z orodjem WinMetr2. Sledi izpis prenesen iz tega programa.

D5.5.2.1 Prva verzija programa

section type and name	start	LOC	LOPC	EL	LC1	LC2	LC3	CM1	CM2
SUMMARY		1172	711	94	120	0.169	155	2055	2.89
\--mod \polinomi\izvorna_koda.sez		1172	711	94	120	0.169	155	2055	2.89
+--mod glavni.cpp		924	591	74	104	0.176	126	1620	2.74
+--r/m ZbrisiZaslon	47	4	4	0	2	0.5	2	1	0.25
+--r/m PolinomJeVpisan	52	7	4	0	1	0.25	1	9	2.25
+--r/m IzpisiPomoc	60	60	42	2	10	0.238	10	173	4.12
+--r/m IzpisiStevilkoZaslona	121	13	8	0	2	0.25	2	14	1.75
+--r/m BeriUporabnikovVnos	135	50	30	3	9	0.3	9	146	4.87
+--r/m IzpisiMeniInVrniIzbiro	186	37	24	2	7	0.292	7	74	3.08
+--r/m IzpisiPolinom	224	20	15	0	6	0.4	6	22	1.47
+--r/m PreberiStevilo	245	71	54	5	10	0.185	10	111	2.06
+--r/m PreberiClen	317	51	31	2	8	0.258	8	116	3.74
+--r/m PretvoriNizVPolinom	369	31	21	0	5	0.238	5	62	2.95
+--r/m IzracunajVrednostPolinoma	401	18	9	0	2	0.222	2	43	4.78
+--r/m BeriPolinomSTipkovnice	420	37	23	2	2	0.087	2	72	3.13
+--r/m BeriPolinomIzDatoteke	458	43	32	5	4	0.125	4	64	2
+--r/m ZbrisiTocke	502	7	4	0	1	0.25	1	8	2
+--r/m IzpisiTocke	510	11	6	0	2	0.333	2	16	2.67
+--r/m DodajTocko	522	20	11	0	2	0.182	2	27	2.45
+--r/m PoisciNicleZBisekcijo	543	58	35	3	12	0.343	12	124	3.54
+--r/m TabelirajPolinom	602	26	12	0	4	0.333	4	64	5.33
+--r/m PreracunajKoordinate	629	35	19	0	3	0.158	3	99	5.21
+--r/m NarisiGrafPolinoma	665	97	72	5	15	0.208	15	154	2.14
+--r/m OmogociInteraktivniIzrisGrafa	763	39	23	4	4	0.174	4	52	2.26
+--r/m ObdelajPolinom	805	52	37	6	6	0.162	6	35	0.946
\--r/m main	858	66	51	5	9	0.176	9	64	1.25
+--mod risar.h		38	18	5	0	0	0	32	1.78
\--mod risar.cpp		210	102	15	18	0.176	29	403	3.95
+--r/m VrniPredznak	18	12	8	0	2	0.25	2	21	2.63
+--r/m Abs	31	11	8	0	2	0.25	2	8	1
+--r/m Round	43	8	4	0	1	0.25	1	15	3.75
+--r/m IntSwap	52	8	5	0	1	0.2	1	10	2
+--r/m DoubleSwap	61	8	5	0	1	0.2	1	10	2
+--r/m PripraviSliko	76	14	6	0	1	0.167	1	33	5.5
+--r/m PrikaziSliko	91	25	13	0	7	0.538	7	84	6.46
+--r/m ZbrisiSliko	117	11	8	0	3	0.375	3	12	1.5
+--r/m NarisiTocko	129	14	6	0	2	0.333	2	30	5

		+-r/m RisiCrto		144	15	11	0	2	0.182	2	8	0.727
		+-r/m NarisiCrto		160	19	7	0	2	0.286	2	49	7
		\-r/m NapisiBesedilo		180	30	17	0	5	0.294	5	72	4.24
-----+-----+-----+-----+-----+-----+-----+												

start ... start line of code section
LOC ... number of all lines in code section
LOPC ... number of program code lines (non-empty, non-comment) in code section
EL ... empty lines in section
LC1 ... logical complexity of section (number of basic testing paths in code section)
LC2 ... logical complexity per line of program code in section (LC1/LOPC)
LC3 ... McCabe's logical complexity
CM1 ... number of words of comment in area
CM2 ... number of words of comment per line of program code (LOPC) in section

Sections:

r/m ... subroutine or method
cls ... class
mod ... module

Done.

D5.5.2.2 Druga verzija programa

section type and name	start	LOC	LOPC	EL	LC1	LC2	LC3	CM1	CM2
SUMMARY		1257	753	96	144	0.191	179	2306	3.06
\--mod \polinomi\izvorna_koda.sez		1257	753	96	144	0.191	179	2306	3.06
+--mod glavni.cpp		1002	633	76	128	0.202	150	1834	2.9
+--r/m ZbrisiZaslon	59	4	4	0	2	0.5	2	1	0.25
+--r/m PolinomJeVpisan	64	10	4	0	1	0.25	1	13	3.25
+--r/m IzpisiPomoc	75	60	42	2	10	0.238	10	173	4.12
+--r/m IzpisiStevilkoZaslona	136	15	8	0	2	0.25	2	21	2.63
+--r/m BeriUporabnikovVnos	152	50	30	3	9	0.3	9	146	4.87
+--r/m IzpisiMeniInVrniIzburo	203	40	24	2	7	0.292	7	78	3.25
+--r/m IzpisiPolinom	244	20	15	0	6	0.4	6	22	1.47
+--r/m PreberiStevilo	265	71	54	5	10	0.185	10	111	2.06
+--r/m PreberiClen	337	58	36	3	10	0.278	10	136	3.78
+--r/m PretvoriNizVPolinom	396	32	21	0	5	0.238	5	73	3.48
+--r/m IzracunajVrednostPolinoma	429	18	9	0	2	0.222	2	43	4.78
+--r/m BeriPolinomSTipkovnice	448	38	23	2	2	0.087	2	74	3.22
+--r/m BeriPolinomIzDatoteke	487	47	32	5	4	0.125	4	79	2.47
+--r/m ZbrisiTocke	535	7	4	0	1	0.25	1	8	2
+--r/m IzpisiTocke	543	11	6	0	2	0.333	2	16	2.67
+--r/m DodajTocko	555	28	18	0	5	0.278	5	40	2.22
+--r/m PoisciNicleZBisekcijo	584	73	47	4	19	0.404	19	143	3.04
+--r/m TabelirajPolinom	658	26	12	0	4	0.333	4	64	5.33
+--r/m PreracunajKoordinate	685	35	19	0	3	0.158	3	99	5.21
+--r/m NarisiGrafPolinoma	721	99	72	5	15	0.208	15	162	2.25
+--r/m OmogociInteraktivniIzrisGrafa	821	48	32	4	12	0.375	12	56	1.75
+--r/m ObdelajPolinom	872	63	44	6	10	0.227	10	63	1.43
\--r/m main	936	66	51	5	9	0.176	9	64	1.25
+--mod risar.h		38	18	5	0	0	0	32	1.78
\--mod risar.cpp		217	102	15	18	0.176	29	440	4.31
+--r/m VrniPredznak	25	12	8	0	2	0.25	2	21	2.63
+--r/m Abs	38	11	8	0	2	0.25	2	8	1
+--r/m Round	50	8	4	0	1	0.25	1	15	3.75
+--r/m IntSwap	59	8	5	0	1	0.2	1	10	2
+--r/m DoubleSwap	68	8	5	0	1	0.2	1	10	2
+--r/m PripraviSliko	83	14	6	0	1	0.167	1	33	5.5
+--r/m PrikaziSliko	98	25	13	0	7	0.538	7	84	6.46
+--r/m ZbrisiSliko	124	11	8	0	3	0.375	3	12	1.5
+--r/m NarisiTocko	136	14	6	0	2	0.333	2	30	5
+--r/m RisiCrto	151	15	11	0	2	0.182	2	8	0.727
+--r/m NarisiCrto	167	19	7	0	2	0.286	2	49	7
\--r/m NapisiBesedilo	187	30	17	0	5	0.294	5	72	4.24

D5.5.3 Podatki o doseženi pokritosti izvirne kode¹:

=====
Pokritost modulov

Ime modula:	Test st.	Zadnji test			Vsi testi		
		ODL	POG	RUT	ODL	POG	RUT
glavni.cpp	19	39.20%	36.01%	90.48%	79.60%	77.38%	100.00%
risar.cpp	19	70.59%	69.05%	91.67%	70.59%	69.05%	91.67%
Skupaj:		42.96%	39.68%	97.14%	78.52%	76.46%	97.14%

=====

¹ Priloži, če si uporabil ustrezno programsko podporo.

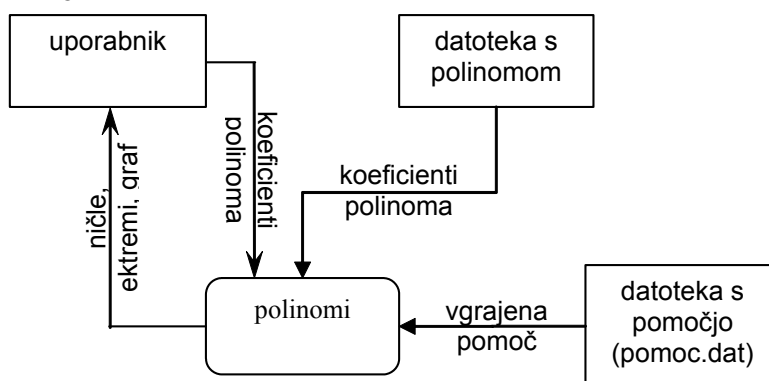
D6 NAČRTOVALSKA DOKUMENTACIJA

D6.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument Načrtovalska dokumentacija	
verzija	1
datum	7.4.2002
lokacija	datoteka Projekt_Polinomi.doc v podmapi »dokumentacija« mape projekta
Odgovorna oseba	
	Miha Štukovnik
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve
	Plan projekta, V0.9
	Sistemske specifikacije V 1.0
	Uporabniški priročnik V 1.0
	Testni primeri 1.0

D6.2 POVZETEK IZ SPECIFIKACIJ

D6.2.1 Kontekstni nivo



Slika 4 - kontekstni nivo

D6.2.2 Datoteke, ki jih uporablja uporabnik

Uporabnik za delovanje programa potrebuje naslednje datoteke:

- polinomi.exe (datoteka z izvršljivo verzijo programa)
- pomoc.dat (datoteka z besedilom vgrajene pomoči)

Uporabnik lahko uporablja tudi neomejeno število tekstovnih datotek, v katerih je s tekstovnim urejevalnikom pripravil polinome. Program omogoča branje polinoma iz teh datotek.

D6.2.3 Zagon programa

Program zaženemo iz ukazne vrstice. Vtipkamo njegovo ime (polinomi). Zaženemo ga lahko tudi iz okenskega raziskovalca z dvoklikom na njegovo ikono ali bližnjico.

D6.2.4 Datoteke, ki jih potrebuje vzdrževalec

Vzdrževalec potrebuje naslednje datoteke:

- celotna projektna dokumentacija (projekt_polinomi.doc)
- polinomi.dsw in polinomi.dsp (datoteki z opisom projekta za Visual Studio 6, ki ju je sicer brez težav mogoče ponovno ustvariti, za prevajanje v drugačnem okolju pa nista potrebni)

op	opcija	
niz	znakovni niz	
Nic	tabela ničel	
Tab	tabela vrednosti	
x		
y		

D6.5 OPISI POSAMEZNIH MODULOV

D6.5.1 Modul ObdelajPolinom

Modul na vhodu sprejme polinom, potem pa od uporabnika pridobi potrebne dodatne podatke (interval iskanja, natančnost, interval risanja) in ustrezno tem podatkov sproži tabeliranje polinoma, iskanje ničel z izbrano natančnostjo ter izris grafa.

D6.5.2 Modul BeriPolinomSTipkovnice

Modul omogoči interaktivni vnos polinoma. Po vnosu vpisani niz pretvori v interni zapis polinoma, preveri veljavnost in vrne polinom (če je bil vnos veljaven).

D6.5.3 Modul BeriPolinomIzDatoteke

Modul omogoča branje polinoma iz datoteke. Od uporabnika pridobi ime datoteke. To datoteko odpre (če seveda obstaja) iz nje prebere vrstico, pretvori niz v polinom in ga vrne.

D6.5.4 Modul IzpisiMeniInVrnilzbiro

Modul izpiše glavni meni programa, prebere uporabnikovo izbiro (zavrne neveljavne možnosti) in vrne izbiro.

D6.5.5 Modul PoisciNicleZBisekcijo

Modul (po)išče ničle podanega polinoma na podanem intervalu s podano natančnostjo. Prva stvar, ki se je modul loti, je razdelitev izbranega intervala na podintervale. To opravi tako, da interval razdeli na dva dela, potem pa rekurzivno vsakega izmed podintervalov spet na dva dela. To stori tolikokrat, kot je določeno s parametrom »slepa globina«. Privzeta vrednost za slepo globino je 19, kar je prilagojeno želeni hitrosti delovanja na priporočeni konfiguraciji. Pri tej slepi globini modul interval iskanja ničel razdeli na 2^{19} (524288) podintervalčkov. Vsako mejo tega intervala preveri, če ni slučajno vrednost funkcije pri izbranem X kar 0. Če je, doda točko v seznam ničel. Sicer za vsak tak podintervalček preveri predznak funkcije na spodnji in zgornji meji intervala. Če se predznak razlikuje, je to dokaz, da je na tem intervalu (če je seveda funkcija zvezna, kar pa polinom gotovo je) vsaj ena ničla. Vse take intervale program še naprej deli, dokler ne zagotovi zahtevane natančnosti iskanja ničle. Ko je ta natančnost zagotovljena, doda točko v seznam ničel. Na koncu dela modul vrne seznam vseh najdenih ničel.

D6.5.6 Modul OmogočiInteraktivnilzrisGrafa

Modul na vhodu sprejme polinom in meje po X in Y osi. Izriše graf na tem območju, potem pa omogoči spreminjanje mej in vnovični izris grafa. In tako, dokler se uporabnik ne odloči za vrnitev v glavni meni.

D6.5.7 Modul TabelirajPolinom

Modul podani polinom tabelira na podanem intervalu s podanim korakom po X osi. Vrne seznam tabeliranih točk.

D6.5.8 Modul PretvoriNizVPolinom

Modul poskrbi za pretvorbo niza v polinom. To je potrebno, ker niz, ki ga program dobi s tipkovnice ali iz datoteke ni primeren za obdelavo. Modul polinom pretvori v interni zapis, kjer je vsak polinom sestavljen s tabelo posameznih členov (stopnje, koeficienti) – glej podatkovni tok P (polinom). Pri tej pretvorbi izvrši tudi preverjanje, če podani niz sploh opisuje veljavni polinom. Če ga, modul vrne polinom v normalizirani obliki (vsak člen druge stopnje)...

D6.5.9 Modul IzračunajVrednostPolinoma

Modul izračuna vrednost podanega polinoma v podani točki. To stori tako, da s pomočjo sistemske knjižnice math (funkcija pow) izračuna vrednost vsakega posameznega člena v podani točki, potem pa vrednosti posameznih členov v tej točki sešteje. Izračunano vrednost vrne kot rezultat.

D6.5.10 Modul Risar

Modul skrbi za izris grafa v psevdografičnem načinu na zaslon. Omogoča pozicionirani izpis besedila, risanje točk in daljic, ter prikaz cele oziroma dela risbe. Risbo zapisuje v posebno podatkovno strukturo, tako da enkrat generirano risbo brez težav večkrat prikaže (na različne načine). Modul je napisan dovolj splošno, da je ponovno uporaben tudi v okviru drugih programov.

D6.5.11 Modul Math

To je standardni zunanji modul programskega jezika C z osnovnimi matematičnimi funkcijami. Uporabljamo ga za izračun potenc pri izračunu vrednosti polinoma v podani točki.

D6.5.12 Modul Stdio

To je standardni zunanji vhodno/izhodni modul programskega jezika C. Omogoča pisanje na zaslon in datoteke, branje z tipkovnice in iz datoteke, itd..

D6.6 NAJPOMEMBNEJŠI PARAMETRI IN OPISI PODATKOVNIH STRUKTUR

D6.6.1 Sintaksa za zapis vhodnega polinoma

Polinom je sestavljen iz posameznih členov.

polinom::=člen+

Člen je sestavljen iz predznaka, koeficienta, spremenljivke x, znaka ^ in stopnje.

člen::=<predznak><koeficient>?(x(^<stopnja>))??

Posebni pogoji, ki niso jasni iz opisa:

- pri člen polinoma ne potrebuje predznaka
- člen nulte stopnje (konstantni člen) ne potrebuje x^0
- člen prve stopnje ne potrebuje oznake potence (1)

Za primere veljavnih polinomov glej testne primere v poglavju D4.6.

D6.6.2 Struktura vhodne datoteke DATOTEKA_S_POLINOMOM

Ta datoteka je namenjena ne-interaktivni vnos polinoma. To je tekstovna datoteka, ki vsebuje eno samo vrstico. Struktura datoteke:

<polinom>

V datoteki je zapisan polinom v enaki obliki, kot je opisano v prejšnjem poglavju (D6.6.1). V datoteki je zapisan en sam polinom. Za njim (in pred njim) ne sme biti zapisano nič drugega.

D6.6.3 Struktura vhodne datoteke s pomočjo

Pomoč za delo s programom ni vkodirana v sam program, ampak je zapisana v zunanji datoteki, ki ima vedno ime pomoc.dat. Struktura datoteke:

@<številka teme pomoči>

<besedilo pomoči za to temo>

<nadaljevanje besedila pomoči za to temo>

@<številka teme pomoči>

<besedilo pomoči za to temo>

...

Številka teme pomoči, ki sledi znaku @ označuje pričetek nove teme pomoči. Številka teme pomoči mora biti unikatna. Nanjo se sklicuje program, ko zahteva vgrajeno pomoč. Besedilo pomoči za posamezno temo se prične v novi vrstici, takoj za številko teme pomoči. Besedilo pomoči ni omejeno po dolžini, potrebno pa je paziti na širino, saj se sicer skvari videz pripravljenega besedila.

D6.6.4 Parametri pri zagonu programa

Ob zagonu pozna program en sam vhodni parameter. To je »-t« (brez narekovajev). Služi vklopu testnega režima delovanja, v katerem se ob delu s programom izpisujejo dodatni diagnostični podatki (številke zaslonov). Ko program poženemo s tem parametrom, ostane v testnem režimu delovanja do naslednjega zagona.

D6.6.5 Podatkovna struktura za zapis polinoma

```
//struktura za zapis polinoma
typedef struct PolinomStruct
{
    int stevilo_clenov;                //skupno stevilo členov (ne stopnja!)
    double eksponent[MAX_KOEF];      //eksponent tega clena
```

```

        double koeficient[MAX_KOEF];          //koeficient tega clena
    } Polinom;

```

Prvi element strukture je celoštevilaska spremenljivka `stevilo_clenov`, ki pove, koliko členov vsebuje polinom – in s tem, koliko elementov vektorjev eksponent in koeficient vsebuje veljavne vrednosti. Vektorja eksponent in koeficient vsebujeta eksponente in koeficiente posameznih členov. `eksponent[0]` in `koeficient[0]` za prvi člen polinoma, `eksponent[1]` in `koeficient[1]` za drugega itd. Število členov je omejeno s konstanto `MAX_KOEF`, ki je v trenutni različici programa nastavljena na vrednost 500. Eksponent in koeficient sta realni števili zapisani v dvojni natančnosti. Ker program deluje pravilno za polinome, se predvideva, da je eksponent vedno realno število ali 0. Koeficient vključuje tudi predznak posameznega člena.

D6.6.6 Podatkovna struktura za zapis množice točk

```

//struktura za zapis mnozice tock (nicle, tabela)
typedef struct TockeStruct
{
    int stevilo_tock;                //stevilo zapisanih tock
    double x[MAX_KOEF];             //vrednost x
    double y[MAX_KOEF];             //f(x)
} MnozicaTock;

```

Ta struktura se uporablja za zapisa množice točk – konkretno ničel polinoma ali rezultata tabeliranja na določenem intervalu. Spremenljivka `stevilo_tock` določa število veljavnih točk zapisanih v vektorjih `x` in `y`. Ta dva vektorja vsebujeta točke (x,y) . Namesto tega bi bila bolj prilagodljiva uporaba dinamične strukture (denimo seznama), a odločili smo se za preprostejšo možnost, katere slabost je navzgor omejeno število točk v množici. Omejitev je, tako kot pri strukturi za zapis polinoma, določena s konstanto `MAX_KOEF`.

D6.7 NATANČNA IDENTIFIKACIJA UPORABLJENIH ORODIJ IN KNJIŽNIC

- Za prevajanje izvorne kode v izvršilni program smo uporabljali Microsoft Visual Studio verzije 6.0 sp5. Predvidevam, da bo izvorno kodo mogoče prevesti tudi z prihodnjimi različicami tega orodja in ostalimi C++ prevajalniki.
- Za izdelavo dokumentacije smo uporabljali Microsoft Word 2002 iz paketa Microsoft Office XP.
- Kot pomožno orodje za izdelavo testnih primerov smo uporabili matematični program Maple 7.00. Namesto tega bi lahko uporabili kak drug matematični program za izračun ničel funkcij in risanje grafov.
- Za pretvorbo dokumentacije v lažje prenosljiv zapis PDF smo uporabili Adobe Acrobat 5.01.
- Za izdelavo datoteke z vgrajeno pomočjo smo uporabljali beležnico iz Windowsov XP (»Microsoft Notepad – verzija 5.1). V ta namen brez težav uporabimo katerikoli drugi tekstovni urejevalnik.
- Za izračun kompleksnosti izvorne kode je bil uporabljen WinMetr 2.11.0
- Za beleženje pokritosti izvorne kode pri testiranju smo uporabljali InstrWin-99.

D6.8 POSTOPEK POTREBEN ZA USTVARJANJE IZVRŠILNE KODE

V Visual Studiu je potrebno odpreti delovni prostor (»workspace«) `polinomi.dsw`. V meniju »Build« izberemo funkcijo »Set Active Configuration« in izberemo konfiguracijo »*polinomi – Win32 Release*«. Potem v meniju »Build« izberemo funkcijo »Build *polinomi.exe*«. Po uspešno opravljeni izgradnji izvršilne kode le-to najdemo v podimeniku »Release« imenika s projektom (z datoteko `polinomi.dsw`).

V prilogi je datoteka **BuildLog.htm**, iz katere je razvidno, kako je bila ustvarjena izvršilna koda.

D6.9 POJMOVNIK

Build Log

D6.9.1 ---Build started: Project: polinomi, Configuration: Debug|Win32 -

Command Lines

```
Creating temporary file
"d:\pro\projekt_polinomi\Verzija2\polinomi\Debug\RSP000003.rsp" with
contents
[
/Od /D "WIN32" /D "_DEBUG" /D "_CONSOLE" /D "_MBCS" /FD /EHsc /RTC1 /MLd
/YX"stdafx.h" /Fp".\Debug\polinomi.pch" /Fo".\Debug/" /Fd".\Debug/" /W3
/c /ZI /TP
.\risar.cpp
.\glavni.cpp
]
Creating command line "cl.exe
@d:\pro\projekt_polinomi\Verzija2\polinomi\Debug\RSP000003.rsp /nologo"
Creating temporary file
"d:\pro\projekt_polinomi\Verzija2\polinomi\Debug\RSP000004.rsp" with
contents
[
/OUT:".\\Debug\polinomi.exe" /NOLOGO /DEBUG /PDB:".\\Debug\polinomi.pdb"
/SUBSYSTEM:CONSOLE /STACK:2000000 /MACHINE:X86 kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
.\Debug\glavni.obj
.\Debug\risar.obj
]
Creating command line "link.exe
@d:\pro\projekt_polinomi\Verzija2\polinomi\Debug\RSP000004.rsp"
```

Output Window

```
Compiling...
risar.cpp
glavni.cpp
Linking...
```

Results

```
Build log was saved at
"file:///d:/pro/projekt_polinomi/Verzija2/polinomi/Debug/BuildLog.htm"
polinomi - 0 error(s), 0 warning(s)
```

D7 UPORABNIŠKI PRIROČNIK

D7.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z OSTALIMI DOKUMENTI

Dokument Uporabniški priročnik	
verzija	1
datum	6.4.2002
lokacija	datoteka prirocnik_polinomi.doc v podmapi dokumentacija\Priročnik.
Odgovorna oseba	
	Miha Štukovnik
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve
	Sistemske specifikacije V 1.0
	Testni primeri 1.0

Program
POLINOMI
(verzija 2.0)

Uporabniški priročnik
(verzija 1.2)

zadnja sprememba: 20.6.2005

D7.2 NAMEN

Program Polinomi je namenjen izračunu ničel, tabeliranju in risanju grafa polinomov. Uporabnik mora obvladati osnovno delo s PC računalnikom. Poznati mora tudi osnove matematične analize.

D7.3 STROJNE IN PROGRAMSKE ZAHTEVE

Program za delovanje potrebuje vsaj računalnik z 80486 procesorjem z 8 MB pomnilnika. Potreben je operacijski sistem DOS ali Windows. V Windowsih program teče v ukaznem načinu (ne uporablja grafičnega uporabniškega vmesnika). Priporočljiv je hiter računalnik (500MHz procesor ali več).

D7.4 NAMESTITEV IN ZAGON PROGRAMA

Program namestimo tako, da vse datoteke iz mape (oziroma arhiva) Polinomi prekopiramo na poljubno mesto na disk. Program lahko poganjamo tudi neposredno z diskete. Poženemo ga lahko s klikom na ikono programa polinomi.exe (v Windowsih). V DOSu se je potrebno najprej premakniti v imenik, kamor smo program namestili, nakar vpišemo ime programa (polinomi).

Program poženemo tako, da njegovo ime vnesemo brez dodatnih argumentov. S programom delamo interaktivno.

Edina **argumenta ukazne vrstice**, ki ju program pozna, sta:

- `/?` (oz. `-h` oz. `--help`), ki služi za izpis pomoči, ki jo pravkar prebiraš
- `-t` (oz. `--test`), ki vključi testni režim delovanja programa. Ta režim delovanja je namenjen preizkuševalcu programa in ne končnemu uporabniku.

D7.5 NAVODILO ZA UPORABO

Ko program poženemo, se prikaže glavni meni programa. V tem meniju so na voljo vpis polinoma (interaktivni ali branje iz datoteke), pomoč za uporabo programa in izhod iz programa. Možnosti v menijih izbiramo tako, da vpišemo številko funkcije, oziroma zvezdico za pomoč, nakar pritisnemo Enter.

```
-----  
Polinomi, verzija 2.00 , (20020329)  
-----  
=====  
GLAVNI MENI  
=====  
1) Vpis polinoma  
2) Branje polinoma iz datoteke  
*) Pomoc  
0) Izhod  
  
>>
```

Slika 5 - Zaslon po zagonu programa

Pomoč za uporabo programa je na voljo vedno, ko program čaka na uporabnikov vnos. Prikličemo jo tako, da namesto pričakovanega vnosa, vnesemo zvezdico in pritisnemo enter.

Interaktivni vnos polinoma

```

=====
VNOS POLINOMA
=====
Vpisi polinom (*=pomoc)

f(x)=x^2-x-2
f(x)=+x^2.000000-1.000000x-2.000000
=====
GLAVNI MENI
=====
1) Vpis polinoma
2) Branje polinoma iz datoteke
3) Izracun nicel in izris grafa funkcije f(x)=x^2-x-2
*) Pomoc
0) Izhod

>>3
f(x)=f(x)=+x^2.000000-1.000000x-2.000000
min x iskanja (primer: -10) = -10
max x iskanja (primer: 10) = 10
natancnost iskanja nicel (primer: 0.001)=0.0000001

```

Slika 6 - Vnos podatkov za izračun ničel polinoma $f(x)=x^2-x-2$

Ta funkcija omogoča interaktivni vnos polinoma, ki ga bomo obdelovali. In kako vnesemo polinom?

Zapišemo ga podobno, kot smo vajeni iz algebre. Najprej predznak člena, ki ga lahko pri prvem členu izpustimo, če je pozitiven. Temu sledi koeficient. Tudi tega lahko izpustimo, če je enak ena. Sledi spremenljivka (vedno x). Seveda jo lahko izpustimo, če gre za člen ničelne stopnje. Za tem zapišemo stopnjo člena. Ker nam uporabniški vmesnik ne omogoča vpisa potenc na običajen matematični način, si pomagamo z znakom ^. Ta pomeni, da je število, ki mu sledi, potenca.

Nekaj primerov:

- $f(x) = x - 1$ zapišemo kot $f(x) = x - 1$.
- $f(x) = x^2 - 2x + 1$ zapišemo kot $f(x) = x^2 - 2x + 1$
- $f(x) = -3.4x^3 + 7x^2 - 4.25x + \frac{4}{3}$ zapišemo kot $f(x) = -3.4x^3 + 7x^2 - 4.25x + 1.333$
- $f(x) = -2x^{22} + 8x^{21} - 2x + 4$ zapišemo kot $f(x) = -2x^{22} + 8x^{21} - 2x + 4$

Branje polinoma iz datoteke

Ta funkcija omogoča branje polinoma iz vnaprej pripravljene tekstovne datoteke. Tako nam dolgih polinomov ni potrebno vedno znova vnašati. Datoteko pripravimo v tekstovnem urejevalniku in jo shranimo kot navadno (ASCII) besedilo, s poljubno končnico.

Zgled:

datoteka pol_c.txt ima v prvi vrstici:

$x^2 - x - 2$

Ko zahtevamo branje polinoma iz datoteke, računalnik čaka na vnos imena datoteke. Tu na običajni način vpišemo ime, če je potrebno pa tudi pot do datoteke.

Nekaj primerov imen datoteke s polinomom:

- polinom.txt
- p15.dat
- c:\polinomi\pol_c.txt
- ..\podatki\p

Izračun ničel in izris grafa polinoma $f(x)=...$

Ta funkcija se v glavnem meniju pojavi šele, ko smo na enega izmed obeh načinov vnesli polinom. S to funkcijo pričnemo iskanje ničel, tabeliranje in izris grafa izbranega polinoma. Funkcija najprej izpiše polinom, kot je uporabnikov vnos interpretiral računalnik. Potem čaka na vnos zgornje in spodnje meje tabeliranja in iskanja ničel (po X osi). Program išče ničle le znotraj izbranega intervala! Takoj po tem bo računalnik vprašal po natančnosti iskanja ničel. Tu vnesemo največjo napako (v vrednosti x), ki jo računalnik lahko naredi pri iskanju ničel.

```
Tabela polinoma na intervalu [-10.000000, 10.000000] s korakom 1.000000:
f(-10.000000)=108.000000;
f(-9.000000)=88.000000;
f(-8.000000)=70.000000;
f(-7.000000)=54.000000;
f(-6.000000)=40.000000;
f(-5.000000)=28.000000;
f(-4.000000)=18.000000;
f(-3.000000)=10.000000;
f(-2.000000)=4.000000;
f(-1.000000)=0.000000;
f(0.000000)=-2.000000;
f(1.000000)=-2.000000;
f(2.000000)=0.000000;
f(3.000000)=4.000000;
f(4.000000)=10.000000;
f(5.000000)=18.000000;
f(6.000000)=28.000000;
f(7.000000)=40.000000;
f(8.000000)=54.000000;
f(9.000000)=70.000000;
f(10.000000)=88.000000;
Pritisni enter.
```

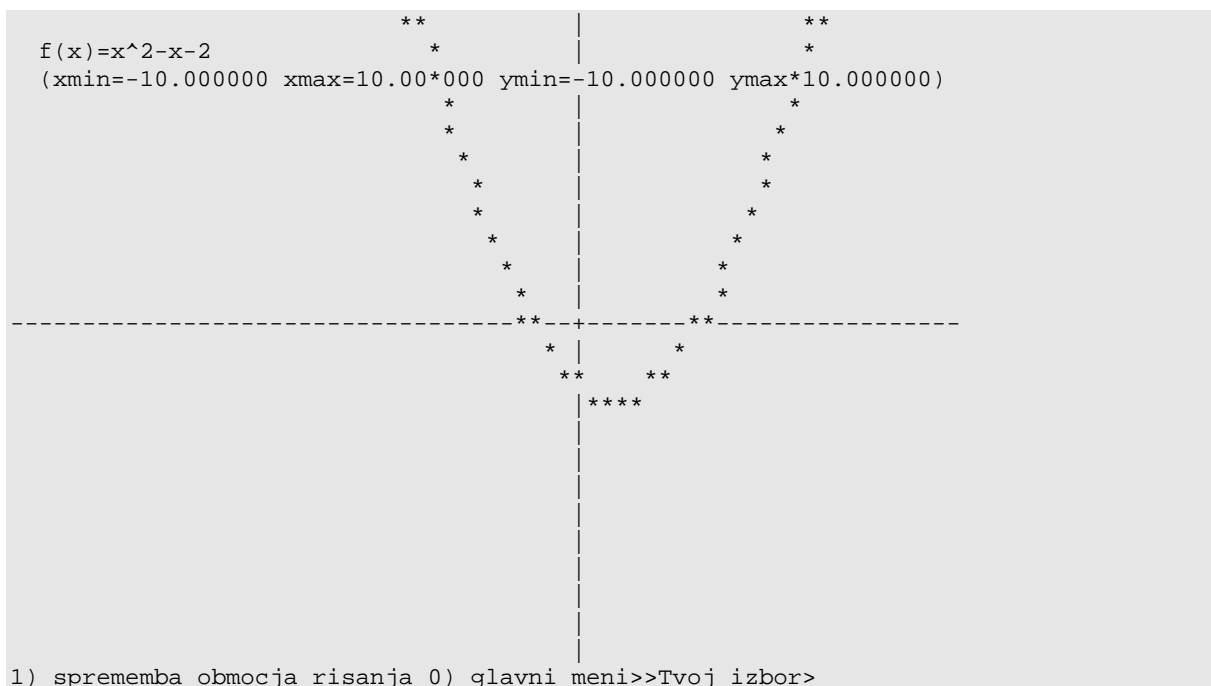
Slika 7 - Izpis tabele polinoma (pri izračunu ničel...) za polinom $f(x)=x^2-x-2$

Izpis izračunanih ničel:

```
Nicle polinoma na intervalu [-10.000000, 10.000000]:
f(-1.000000)=-0.000000;
f(2.000000)=-0.000000;
Pritisni enter.
```

Slika 8 - Izpis najdenih ničel polinoma $f(x)=x^2-x-2$

Nato lahko graf tudi narišemo:



Slika 9 - Izris grafa polinoma $f(x)=x^2-x-2$

Ko je graf izrisan, nam računalnik da v meniju na voljo spremembo območja risanja. Kot pove ime, s to funkcijo spremenimo območje risanja funkcije (po X in Y osi) ter tako po potrebi spremenimo merilo, podrobneje narišemo pomembne dele funkcije ipd. Če izberemo spreminjanje območja, bo računalnik čakal na vnos štirih vrednosti – torej dveh intervalov na X in Y osi, nakar bo graf ponovno izrisal. To lahko ponavljamo, kakor dolgo hočemo.

D7.6 HIBE PROGRAMA

Edina znana hiba druge verzije programa je nesposobnost programa, da bi zanesljivo odkril ničle sodih stopenj (dvojne, četverne...), v katerih se graf polinoma le dotakne osi X. Potencialno se lahko zgodi, da program ne bi našel tudi kake druge ničle, saj uporabljeni algoritem omogoča le zanesljivo iskanje ničle, ki leži med podanim začetkom in koncem intervala, na katerem funkcija spremeni predznak.

D8 IZVORNA KODA – IZPIS

D8.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z OSTALIMI DOKUMENTI

Dokument	
Izvorna koda – izpis	
verzija	1
datum	6.4.2002
lokacija	Datoteke glavni.cpp.pdf, risar.cpp.pdf in risar.h.pdf (obarvana formatirana verzija za izpis) v podimeniku dokumentacija. Datoteke glavni.cpp, risar.cpp in risar.h (originalne tekstovne datoteke) v podimeniku razvoj.
Odgovorna oseba	
	Miha Štukovnik
Reference na drugo dokumentacijo	

D9 IZVRŠILNA KODA

D9.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z OSTALIMI DOKUMENTI

Dokument	
Izvršilna koda	
verzija	1
datum	6.4.2002
lokacija	Datoteka polinomi.exe v osnovnem imeniku projekta polinomi«. Za delovanje je potrebna je tudi datoteka pomoc.dat v istem imeniku
Odgovorna oseba	
	Miha Štukovnik
Reference na drugo dokumentacijo	

D10 OPIS VSEH DATOTEK

D10.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z OSTALIMI DOKUMENTI

Dokument	
Izvršilna koda	
verzija	1
datum	6.4.2002
lokacija	datoteka Projekt_Polinomi.doc (oziroma .pdf) v podmapi »dokumentacija« mape projekta
Odgovorna oseba	
	Miha Štukovnik
Reference na drugo dokumentacijo	

Datoteke projekta Polinomi so v prvem nivoju razdeljene v pet map.

- dokumentacija
- razvoj
- prototip
- verzija1
- verzija2

V mapi dokumentacija se nahaja projektna dokumentacija. Glavni dokument se imenuje Projekt_Polinomi.doc. K njemu spadajo še posebej shranjene slike tp?graf.wmf ter neustreznost_?.tif. V datotekah glavni_cpp.pdf, risar.cpp.pdf ter risar.h.pdf je shranjena formatirana izvorna koda druge verzije programa. V podmapi Priročnik je shranjen uporabniški priročnik za uporabo programa Polinomi.

V mapi »razvoj« je shranjena aktualna (zadnja) verzija projekta – torej vsa izvorna koda, projektne datoteke, preveden program.

Mape prototip, verzija1 in verzija2 so posnetki stanja projekta (izvorna koda, program, dokumentacija), kot so bili aktualni v istoimenski različici programa (mejniku projekta).

Sledi izpis celotne strukture mape projekt_polinomi in dodatna razlaga za posamezne datoteke:

```
Directory of D:\pro\projekt_polinomi
22.04.2002  13:01    <DIR>          dokumentacija
    Projektna dokumentacija vključno z uporabniškim priročnikom.
15.04.2002  10:32             73.728 polinomi.exe
    Program Polinomi verzija 2.0. Končna različica programa prevedena za končnega uporabnika.
12.02.2002  15:28             3.341 pomoc.dat
    Vgrajena pomoč za program polinomi.
01.02.2002  12:59    <DIR>          prototip
    Projekt polinomi – posnetek stanja pri verziji »prototip«.
19.04.2002  14:59    <DIR>          razvoj
    Razvojna verzija programa.
25.02.2002  10:17    <DIR>          Verzija1
    Posnetek stanja pri verziji 1.0.
25.03.2002  09:48    <DIR>          Verzija2
    Posnetek stanja pri verziji 2.0

Directory of D:\pro\projekt_polinomi\dokumentacija
15.04.2002  10:58             72.180 glavni.cpp.pdf
15.04.2002  11:25             42.556 risar.cpp.pdf
15.04.2002  11:25             35.748 risar.h.pdf
    Izvorna koda končne različice programa v formatirani obliki.
10.04.2002  10:19             252.978 neustreznost_1.tif
10.04.2002  12:16             202.814 neustreznost_2.tif
10.04.2002  12:17             208.182 neustreznost_3.tif
10.04.2002  12:18             225.290 neustreznost_4.tif
```

10.04.2002 12:27 215.142 neustreznost_5.tif
10.04.2002 12:33 215.970 neustreznost_6.tif
10.04.2002 12:41 204.798 neustreznost_7.tif
10.04.2002 12:41 221.104 neustreznost_8.tif
10.04.2002 12:42 218.186 neustreznost_9.tif
Skenirane slike poročil o neustreznostih.
08.11.2001 09:58 26.112 polinomi.doc
Originalna datoteka z naročnikovimi zahtevami.
22.04.2002 12:32 <DIR> Priročnik
Podimenik z uporabniškim priročnikom.
22.04.2002 12:49 1.372.672 Projekt_Polinomi.doc
Dokument, ki ga pravkar bereš.
12.04.2002 12:53 424.960 standard.doc
Standard, standard CVVS 2-2000
07.03.2002 13:10 447.118 tp10graf.wmf
07.03.2002 13:18 447.118 tp11graf.wmf
07.03.2002 10:21 447.118 tp1graf.wmf
07.03.2002 10:03 447.118 tp2graf.wmf
07.03.2002 10:25 447.118 tp3graf.wmf
07.03.2002 10:46 447.118 tp4graf.wmf
07.03.2002 10:53 447.118 tp5graf.wmf
07.03.2002 11:11 447.118 tp6graf.wmf
07.03.2002 11:12 447.118 tp7graf.wmf
07.03.2002 11:20 447.118 tp8graf.wmf
07.03.2002 12:03 447.118 tp9graf.wmf
Slike grafov, ki pripadajo testnim primerom.
13.03.2002 12:24 28 tg25.txt
Datoteka pripadajoča testnemu primeru TG2/5.

Directory of D:\pro\projekt_polinomi\dokumentacija\Priročnik
22.04.2002 12:16 39.936 prirocnik_polinomi.doc
22.04.2002 12:12 3.364 slika1.tif
22.04.2002 12:13 4.802 slika2.tif
22.04.2002 12:13 5.728 slika3.tif
22.04.2002 12:13 4.032 slika4.tif
Priročnik za uporabo programa Polinomi in pripadajoče slike.

Directory of D:\pro\projekt_polinomi\prototip
22.04.2002 12:59 <DIR> Debug
Imenik s izvršljivo kodo prototipa.
30.01.2002 10:58 21.135 glavni.cpp
Izvorna koda glavnega programa.
30.11.2001 09:56 826 ostenek.h
Izvorna koda izločena iz projekta.
03.12.2001 14:51 311.808 Projekt_Polinomi.doc
Takratna različica projektne dokumentacije.
01.02.2002 12:59 4.490 prototip.dsp
01.02.2002 11:06 539 prototip.dsw
01.02.2002 12:59 41.984 prototip.ncb
01.02.2002 12:59 48.640 prototip.opt
01.02.2002 11:07 1.709 prototip.plg
Projektne datoteke za Visual Studio 6.
28.01.2002 14:58 1.775 risar.cpp
Modul za risanje.
28.01.2002 14:54 540 risar.h
Glava modula za risanje.

Directory of D:\pro\projekt_polinomi\prototip\Debug
01.02.2002 11:07 225.337 polinomi.exe
Izvršljivi program – prototip.

Directory of D:\pro\projekt_polinomi\razvoj
22.04.2002 12:59 <DIR> Debug
imenik z izvršljivo verzijo programa (način za razhroščevanje)
15.04.2002 10:49 31.695 glavni.cpp
izvorna koda glavnega programa
25.02.2002 10:19 30 izvorna_koda.sez
Seznam datotek z izvorno kodo. To je vhodna datoteka za program za merjenje kompleksnosti WinMetr.
15.04.2002 11:47 4.527 polinomi.dsp
23.11.2001 17:42 539 polinomi.dsw
projektne datoteke za Visual Studio 6
11.02.2002 13:27 233.529 polinomi.exe
izvršljiva verzija programa (razvojna)
19.04.2002 14:59 66.560 polinomi.ncb
19.04.2002 14:59 48.640 polinomi.opt
15.04.2002 10:32 1.266 polinomi.plg
projektne datoteke za Visual Studio 6
12.02.2002 15:28 3.341 pomoc.dat
vgrajena pomoč za program polinomi

22.04.2002 12:59 <DIR> Release
imenik z izvršljivo verzijo programa (optimizirana)

15.04.2002 10:34 5.820 risar.cpp
modul za risanje

15.04.2002 11:24 1.217 risar.h
glava modula za risanje

13.03.2002 12:24 28 tg25.txt
datoteka pripadajoča testnemu primeru TG2/5

11.02.2002 09:28 8 tpl.txt
še ena vhodna datoteka za testiranje

Directory of D:\pro\projekt_polinomi\razvoj\Debug

10.04.2002 14:03 233.529 polinomi.exe
izvršljiva verzija programa (prevedena v testnem režimu)

Directory of D:\pro\projekt_polinomi\razvoj\Release

15.04.2002 10:32 73.728 polinomi.exe
izvršljiva verzija programa (optimizirana)

Directory of D:\pro\projekt_polinomi\Verzija1

11.03.2002 13:31 <DIR> polinomi
imenik z izvorno kodo in datotekami Visual Studia

08.11.2001 09:58 26.112 polinomi.doc
Originalna datoteka z naročnikovimi zahtevami.

21.02.2002 16:19 501.760 Projekt_Polinomi.doc
Projektna dokumentacija. Posnetek pri prvi verziji.

30.01.2002 13:12 447.118 tplgraf.wmf
In pripadajoč graf za testni primer 1.

Directory of D:\pro\projekt_polinomi\Verzija1\polinomi

Datoteke z izvorno kodo in projektne datoteke Visual Studia. Komentarje glej pri opisu razvojnega imenika.

22.04.2002 12:59 <DIR> Debug

20.02.2002 14:29 28.571 glavni.cpp

22.04.2002 13:00 <DIR> Ins

11.03.2002 13:31 842 instrumentacija.ini
Projektna datoteka za orodje InstrWin-99.

25.02.2002 10:19 30 izvorna_koda.sez
Projektna datoteka za orodje WinMetr 2.

05.02.2002 16:04 4.528 polinomi.dsp

23.11.2001 17:42 539 polinomi.dsw

11.02.2002 13:27 233.529 polinomi.exe

25.02.2002 10:17 66.560 polinomi.ncb

25.02.2002 10:17 48.640 polinomi.opt

13.02.2002 11:22 1.299 polinomi.plg

12.02.2002 15:28 3.341 pomoc.dat

11.02.2002 13:12 5.470 risar.cpp

11.02.2002 12:28 1.219 risar.h

22.11.2001 11:56 2 tekst.dat

11.02.2002 09:28 8 tpl.txt

Directory of D:\pro\projekt_polinomi\Verzija1\polinomi\Debug

13.02.2002 11:22 233.529 polinomi.exe
Izvršljiva verzija programa v1.0.

Directory of D:\pro\projekt_polinomi\Verzija1\polinomi\Ins

Imenik z instrumentirano različico programa.

25.02.2002 10:58 784 glavni.cfx
Konfiguracija za preverjanje v instrumentatorju. Generira in vzdžuje jo InstrWin.

25.02.2002 10:58 35.654 glavni.cpp
Instrumentirana različica izvorne kode glavnega programa. Generira in vzdžuje jo InstrWin.

11.03.2002 13:27 42.504 glavni.dnm
»Dinamična« datoteka instrumentatorja (za glavni modul). Generira in vzdžuje jo InstrWin.

25.02.2002 10:58 4.875 glavni.sta
»Statična« datoteka instrumentatorja (za glavni modul). Generira in vzdžuje jo InstrWin.

25.02.2002 11:50 4.561 polinomi.dsp

23.11.2001 17:42 539 polinomi.dsw

Projektni datoteki visual studia. Prekopirani iz imenika nad njim.

25.02.2002 10:59 266.310 polinomi.exe
Izvršljiv instrumentirani program.

11.03.2002 13:05 50.176 polinomi.ncb

11.03.2002 13:05 48.640 polinomi.opt

25.02.2002 10:59 1.537 polinomi.plg

Projektna datoteka visual studia.

12.02.2002 15:28 3.341 pomoc.dat
Vgrajena pomoč.

12.04.2002 09:43 11.016 report.rpt
Poročilo programa InstrWin.

25.02.2002 10:58 7.191 risar.cpp
Instrumentirana različica izvorne kode modula risar. Generira in vzdžuje jo InstrWin.

```

11.03.2002 13:27          7.644 risar.dnm
      Dinamična datoteka za modul risar. Generira in vzdžuje jo InstrWin.
11.02.2002 12:28          1.219 risar.h
      Glava modula risar. Prekopirana iz imenika više.
25.02.2002 10:58          1.048 risar.sta
      Statična datoteka modula risar.
11.03.2002 13:27          6.604 stat.log
      Dnevnik zagonov instrumentiranega programa. Generira in vzdžuje jo InstrWin.
25.02.2002 10:58          89 test.ins
      Konfiguracija za instrumentirani program. Generira in vzdžuje jo InstrWin.
11.02.2002 09:28          8 tpl.txt
      Vhodna datoteka za preverjanje.

```

Directory of D:\pro\projekt_polinomi\Verzija2

Glej opise datotek pri verziji 1.

```

25.03.2002 11:57      <DIR>          polinomi
08.11.2001 09:58          26.112 polinomi.doc
11.03.2002 13:31          1.065.472 Projekt_Polinomi.doc

```

Directory of D:\pro\projekt_polinomi\Verzija2\polinomi

Glej opise datotek pri verziji 1\polinomi.

```

15.04.2002 10:49          31.695 glavni.cpp
22.04.2002 13:00      <DIR>          ins
25.03.2002 11:52          842 instrumentacija.tpr
25.02.2002 10:19          30 izvorna_koda.sez
13.03.2002 16:01          4.589 polinomi.dsp
23.11.2001 17:42          539 polinomi.dsw
21.03.2002 14:25          73.728 polinomi.exe
25.03.2002 11:57          66.560 polinomi.ncb
25.03.2002 11:57          48.640 polinomi.opt
25.03.2002 11:56          1.332 polinomi.plg
12.02.2002 15:28          3.341 pomoc.dat
22.04.2002 13:00      <DIR>          Release
15.04.2002 10:34          5.820 risar.cpp
15.04.2002 11:24          1.217 risar.h
22.11.2001 11:56          2 tekst.dat
13.03.2002 12:24          28 tg25.txt
11.02.2002 09:28          8 tpl.txt

```

Directory of D:\pro\projekt_polinomi\Verzija2\polinomi\ins

Glej opise datotek pri verziji 1\polinomi\ins.

```

25.03.2002 11:57          784 glavni.cfx
18.04.2002 12:28          39.998 glavni.cpp
18.04.2002 12:28          5.774 glavni.sta
18.04.2002 12:28          784 polinomi.cfx
25.03.2002 11:56          4.683 polinomi.dsp
23.11.2001 17:42          539 polinomi.dsw
25.03.2002 11:57          274.502 polinomi.exe
25.03.2002 11:57          50.176 polinomi.ncb
25.03.2002 11:57          48.640 polinomi.opt
25.03.2002 11:57          1.541 polinomi.plg
12.02.2002 15:28          3.341 pomoc.dat
18.04.2002 12:28          992 report.rpt
18.04.2002 12:28          7.541 risar.cpp
11.02.2002 12:28          1.219 risar.h
18.04.2002 12:28          1.048 risar.sta
25.03.2002 11:57          3.427 stat.log
25.03.2002 10:31          89 test.ins

```

Directory of D:\pro\projekt_polinomi\Verzija2\polinomi\Release

```

25.03.2002 11:56          73.728 polinomi.exe

```

Izvršljiva optimizirana različica druge verzije programa polinomi. Namenjena končnemu uporabniku. Program v osnovnem imeniku projekta je enak.