
Interaktivní Smithův diagram

X33TSW – Testování a diagnostika softwaru

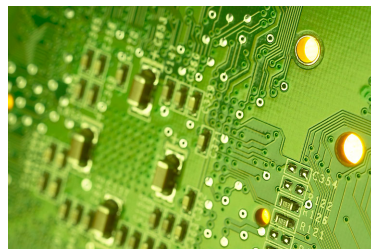
Helena Blažková & Roman Mikulec (ST 16:15 - 17:45)

Fakulta elektrotechnická • ČVUT v Praze • 28. května 2008

ZADÁNÍ

1. Vytvořte dokument obsahující systémové a softwarové požadavky týkající se vašeho projektu. Znázorněte jejich vazby. Doporučujeme použít program RequisitePro.
2. Vytvořte UML model testovaného softwarového projektu (class diagram a sequence diagram). Doporučujeme použít program Rose.
3. Vytvořte diagram příčin a účinků (CE diagram) vzhledem k parametrům ovlivňující výkonnost a funkci testovaného programu.
4. Zjistěte, zda testovaný kód neobsahuje chyby ve správě paměti. Doporučujeme použít program Purify. Pokud by program takové problémy náhodou neobsahoval, zkuste je uměle vyvolat.
5. Zjistěte, zda by testovaný kód nemohl být výkonnější. Doporučujeme použít program Quantify.
6. Zjistěte, zda se ve vašem kódu nevyskytují části, které nejsou nikdy volány. Doporučujeme použít program PureCoverage.
7. Manuální testování s pomoci Rational Manual Tester
8. Navrhněte a implementujte testovací skripty grafického rozhraní testovaného softwaru. Doporučujeme použít program Rational Robot.
9. Na vybraném problému (vhodnost konzultujte s cvičícím) odvodte vztah popisující šíření chyb. Pokuste se jej ověřit.

Obsah



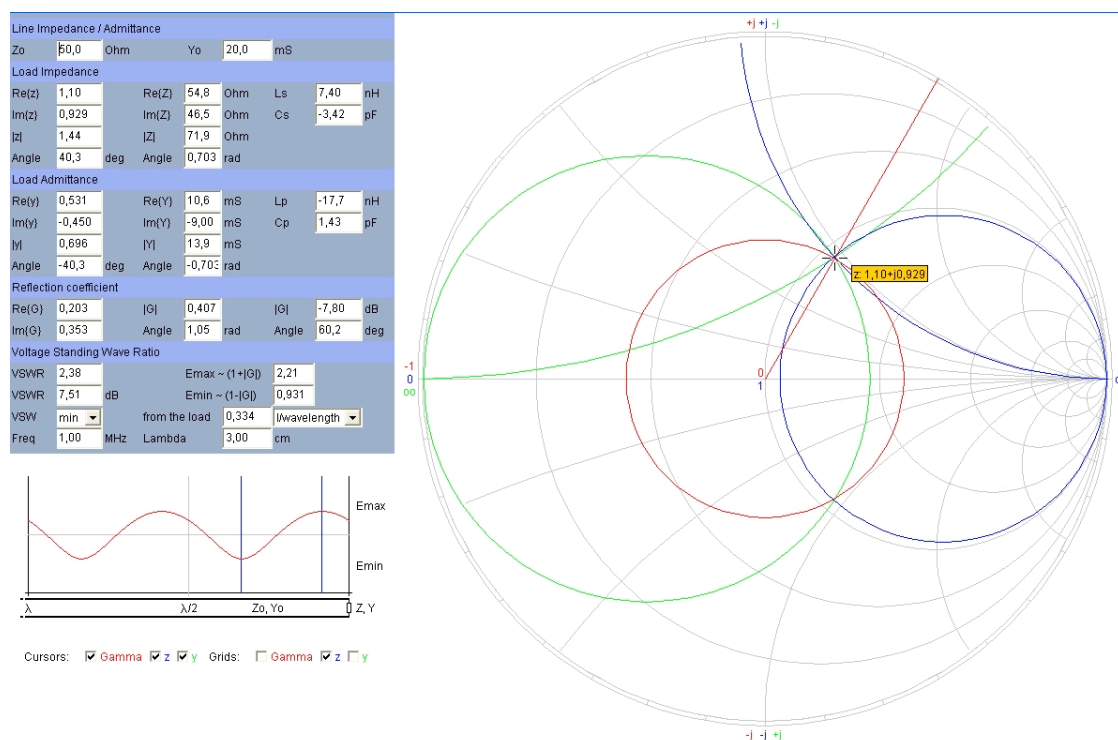
Úvod	3
Popis programu	3
Ovládání programu	4
Systémové a softwarové požadavky	5
Systémové požadavky	5
Funkční požadavky	5
Traceability matrix	7
UML model	8
Class diagram	9
Sequence diagram	10
Diagram příčin a účinků	11
Chyby ve správě paměti	12
Zhodnocení	13
Výkonnější kód	14
Zhodnocení	15
Nevolané části programu	16
Manuální testování	18
Zhodnocení	19
Testování grafického prostředí	20
Reakce na nestandardní vstup	20
Správnost výpočtu	24
Zhodnocení	26
Šíření chyb	27
Závěr	29

Úvod

Popis zpracovávaného programu Interaktivní Smithův diagram

POPIS PROGRAMU

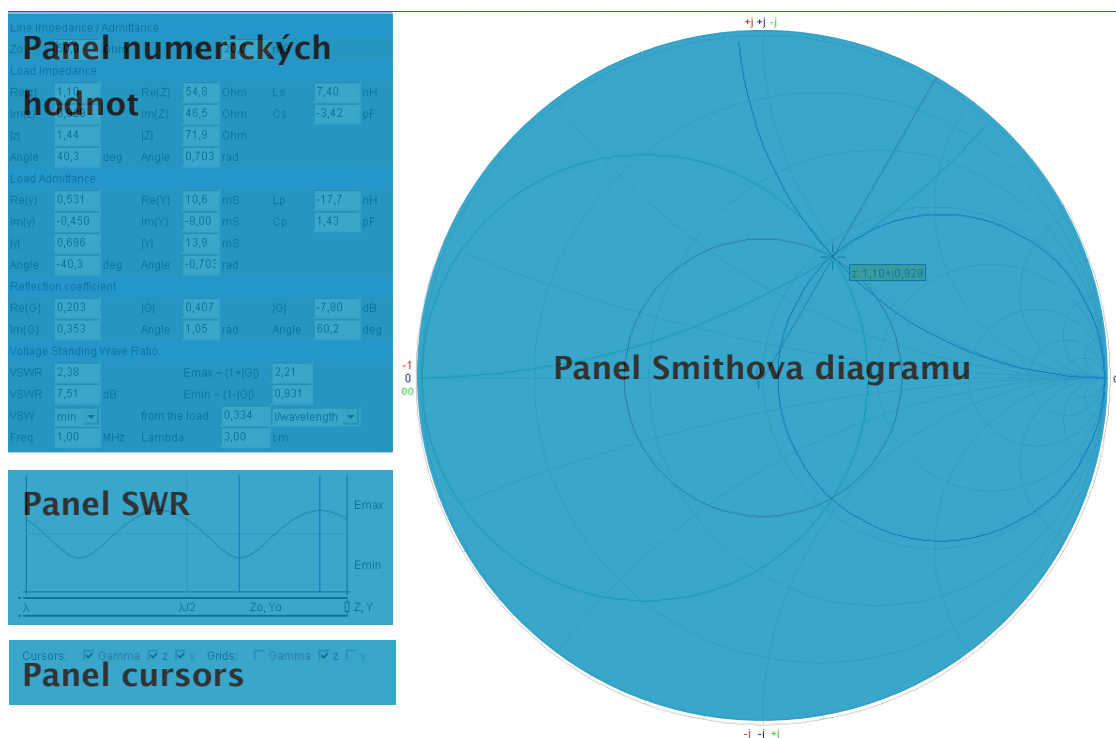
Interaktivní Smithův diagram vznikl jako učební pomůcka na ČVUT – FEL. Jeho hlavním účelem je grafická a numerická reprezentace vybraných parametrů vedení se stojatým vlněním.



Ukázka pracovního okna programu

Z hlediska uživatele programu jej lze rozdělit do 4 základních částí:

1. Panel numerických hodnot – panel je určen k přímému vkládání numerických hodnot
2. Panel SWR – zobrazuje stojaté vlnění, tento panel není interaktivní
3. Panel Smithova diagramu – interaktivní Smithův diagram
4. Panel cursors – umožňuje zapínat nebo vypínat zobrazení souřadných systémů a jednotlivých parametrů na Smithově diagramu.



Jednotlivé části programu

OVLÁDÁNÍ PROGRAMU

Ovládání programu je velmi jednoduché. Na začátku stačí zadat Z_0 a potom můžeme editovat jednotlivé parametry buď zapsáním přesné numerické hodnoty, a potvrzením klávesou enter, nebo kliknutím na zvolený bod přímo ve Smithově diagramu.

Provedené změny se promítnou do všech dalších částí programu.

Panel numerických hodnot je určen k přímému vkládání numerických hodnot. Můžete vložit jakoukoliv hodnotu, vnitřní kontrola upraví vložená data do správného rozsahu a formátu. Přepočítané hodnoty a aktualizovaný diagram se okamžitě zobrazí. Zobrazené hodnoty jsou typicky zaokrouhlené na 3 desetinná místa.

Panel SWR je situován pod panelem numerických hodnot. Červená křivka zobrazuje stojaté vlnění, modré čáry označují minimum a maximum stojatého vlnění. Tento panel není interaktivní.

Panel Smithova diagramu se ovládá myší. Pokud kliknete do diagramu, přesune se do tohoto místa kurzor, a překreslí se ukazatele impedance, admittance a SWR.

Můžete také kliknout myší a přesouvat ukazatel přes diagram, stále se budou překreslovat a přepočítávat aktuální parametry. Checkboxy vlevo dole zapínají/vypínají zobrazování jednotlivých souřadných systémů a ukazatele hodnot.

Systémové a softwarové požadavky

Základním krokem tvorby nového softwaru je stanovení požadavků na něj kladených. Ty lze rozdělit na požadavky funkční a systémové. Mezi nimi existují vzájemné vazby. Ty lze popsat pomocí traceability matrix.

V našem případě jsme ke tvorbě požadavků a příslušné traceability matrix použili program RequisitePro. Výsledek je v upravené podobě zobrazen níže.

SYSTÉMOVÉ POŽADAVKY

1. JRE > 1.5 – Je vyžadováno běhové prostředí JRE alespoň ve verzi 1.5
2. Grafické rozhraní OS
3. Webový prohlížeč
4. Java plugin pro prohlížeč

FUNKČNÍ POŽADAVKY

1. Funkční požadavky
 - 1.1. Výpočet PSW
 - 1.2. Výpočet Gama
 - 1.3. Výpočet přizpůsobovacích impedancí
 - 1.4. Výpočet admitance
 - 1.5. Dopočítání ostatních parametrů
 - 1.6. Výpočet impedance
 - 1.7. Převod úhlových jednotek
 - 1.8. Rozklad na reálnou a imaginární složku
 - 1.9. Zobrazení vypočtených hodnot

2. Uživatelské rozhraní

2.1. Grafické rozhraní – Je požadováno grafické rozhraní

2.1.1. GUI aplikace – Je požadován běh ve formě aplikace

2.1.2. Java applet – Je požadován běh ve formě Java appletu

2.1.3. Vykreslení Smithova diagramu – Program dokáže vykreslit data v podobě Smithova diagramu

2.1.4. Výběr zobrazených os ve Smithově diagramu

2.1.5. Zobrazení PSW panelu

2.2. Zadávání dat

2.2.1. Přímé zadávání dat – Uživatel může zadat data přímo pomocí textboxů a comboboxů

2.2.2. Zadávání pomocí Smithova diagramu

2.2.3. Kontrola hodnot zadaných uživatelem

2.2.4. Převod uživatelem zadaných dat do vhodného formátu

TRACEABILITY MATRIX

	SYS1: JRE > 1.5	SYS2: Grafické rozhraní OS	SYS3: Webový prohlížeč	SYS4: Java plugin pro prohlížeč
<input type="checkbox"/> FEAT1: Funkční požadavky				
FEAT1.1: Výpočet PSW				
FEAT1.2: Výpočet Gama				
FEAT1.3: Výpočet přizpůsobovacích impedancí				
FEAT1.4: Výpočet admitance				
FEAT1.5: Dopočítání ostatních parametrů				
FEAT1.6: Výpočet impedance				
FEAT1.7: Převod úhlových jednotek				
FEAT1.8: Rozklad na reálnou a imaginární složku				
FEAT1.9: Zobrazení vypočtených hodnot		↗		
<input type="checkbox"/> FEAT2: Uživatelské rozhraní				
<input type="checkbox"/> FEAT2.1: Grafické rozhraní				
FEAT2.1.1: GUI aplikace	↗	↗		
FEAT2.1.2: Java applet	↗	↗	↗	↗
FEAT2.1.3: Vykreslení Smithova diagramu		↖		
FEAT2.1.4: Výběr zobrazených os ve Smithově diagramu		↖		
FEAT2.1.5: Zobrazení PSW panelu		↖		
<input type="checkbox"/> FEAT2.2: Zadáání dat				
FEAT2.2.1: Přímé zadávání dat		↖		
FEAT2.2.2: Zadáání pomocí Smithova diagramu		↖		
FEAT2.2.3: Kontrola hodnot zadaných uživatelem	↗			
FEAT2.2.4: Převod uživatelem zadaných dat do vhodného formátu	↗			

UML model

Unified Modeling Language (UML) je prostředek k vizualizaci, specifikaci a navrhování používaný v objektově orientovaném programování.

V současnosti se používají dvě základní verze tohoto jazyka. Starší verze 1 a novější verze 2, která se od předešlé mírně odlišuje.

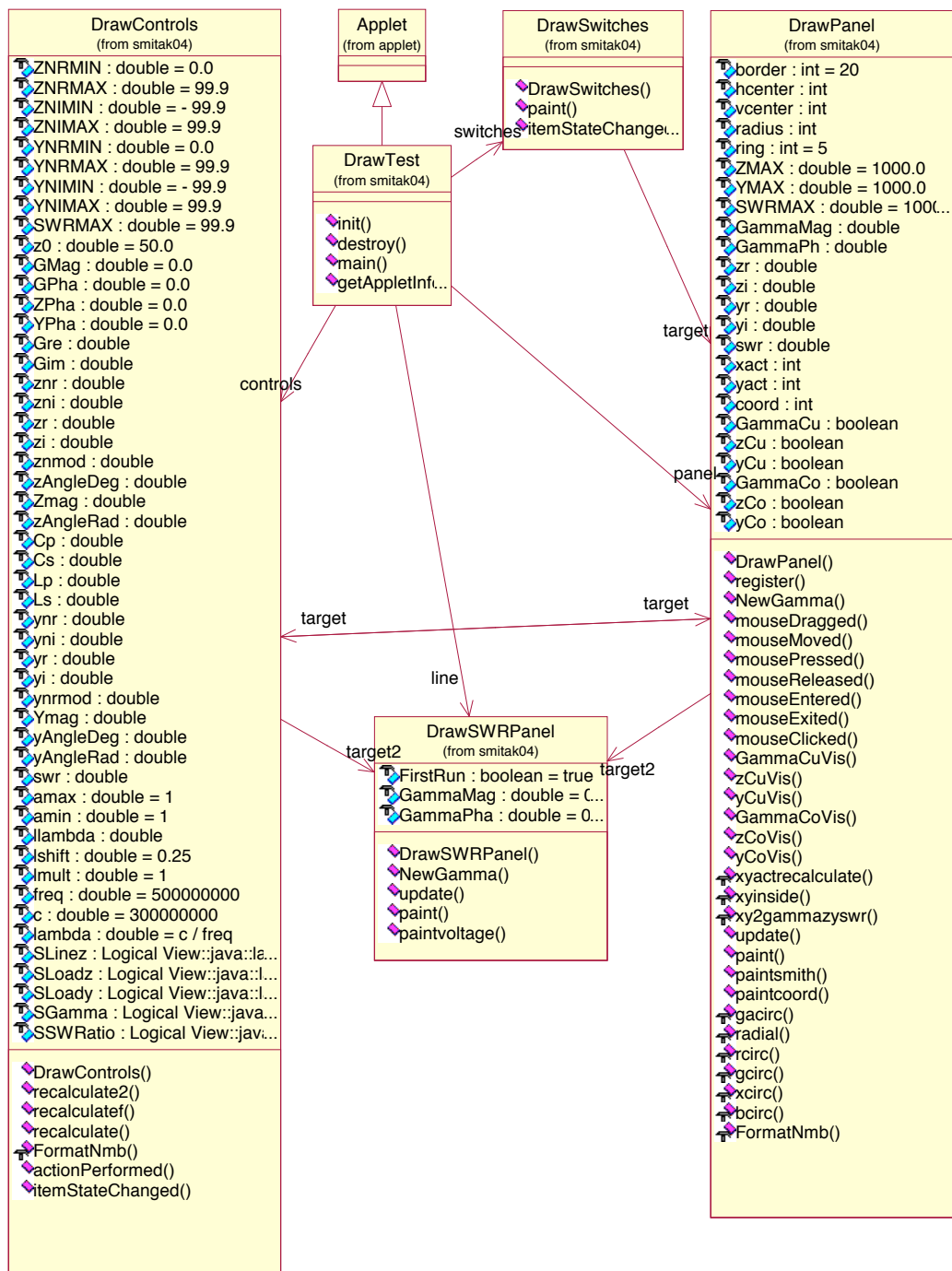
Diagramy v UML lze rozdělit na:

- strukturní (structure)
 - tříd (class)
 - komponent (component)
 - composite structure
 - nasazení (deployment)
 - objektů (object)
 - balíčků (package)
- chování (behavior)
 - aktivit (activity)
 - stavový (state machine)
 - užití (use case)
- interakce (interaction)
 - komunikační (communication)
 - interaction overview
 - sekvenční (sequence)
 - časování (timing)

V našem případě přikládáme ukázkou class diagramu a sequence diagramu.

K jejich vytvoření jsme použili program Rational Rose, do kterého jsme naimportovali existující zdrojové kódy programu.

CLASS DIAGRAM

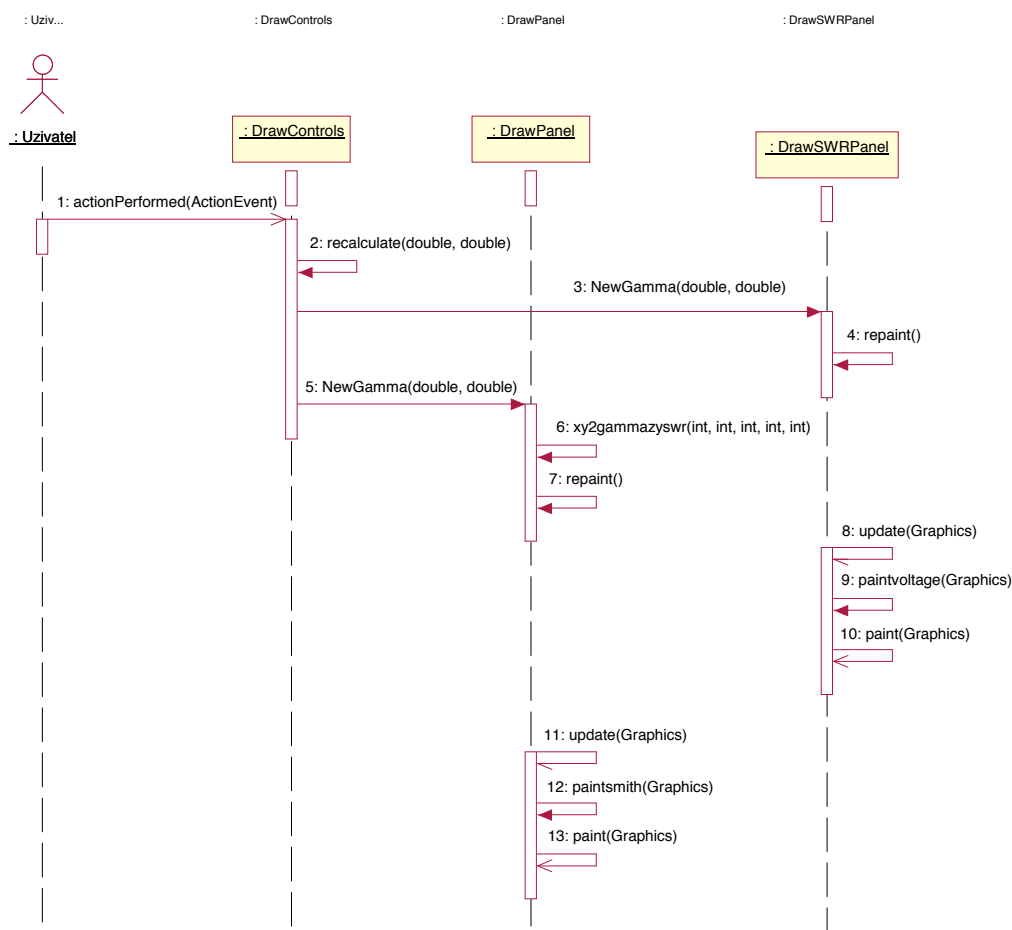


Class diagram aplikace

Hlavní třída **DrawTest** je vytvořen jako applet a vznikla tedy děděním od třídy **Applet**. V jejím rámci se nacházejí třídy **DrawControls**, **DrawSwitches**, **DrawSWRPanel** a **DrawPanel**. Jejich význam odpovídá členění aplikace uvedeném v rámci jejího popisu.

Jednotlivé třídy si uchovávají navzájem odkazy na další třídy. V případě změny hodnoty v jedné třídě jsou pak vyvolány změny ve zbylých třídách.

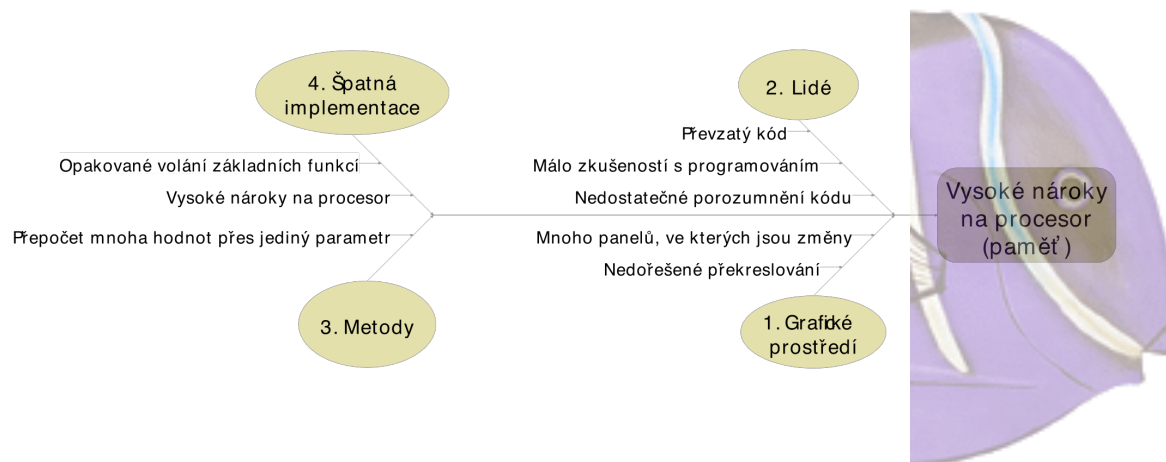
SEQUENCE DIAGRAM



Sequence diagram při změně zadaných hodnot

Sequence diagram zachycuje situaci, kdy uživatel změnil hodnotu v panelu numerických hodnot (DrawControls). Následně jsou přepočteny hodnoty uvnitř panelu. Poté je vynuceno přepočtení hodnot i v dalších panelech (DrawPanels – Smithův diagram, DrawSWRPanel). Tam jsou hodnoty přepočteny a je zavoláno jejich překreslení.

Diagram příčin a účinků



CE diagram

Chyby ve správě paměti

Náš testovaný program je napsán v programovacím jazyce JAVA. Ten není vzhledem ke svým principům náchylný ke vzniku tzv. memory leaks jak je známe z jazyků jako c, c++. O uvolnění se zde stará garbage collector (GC).

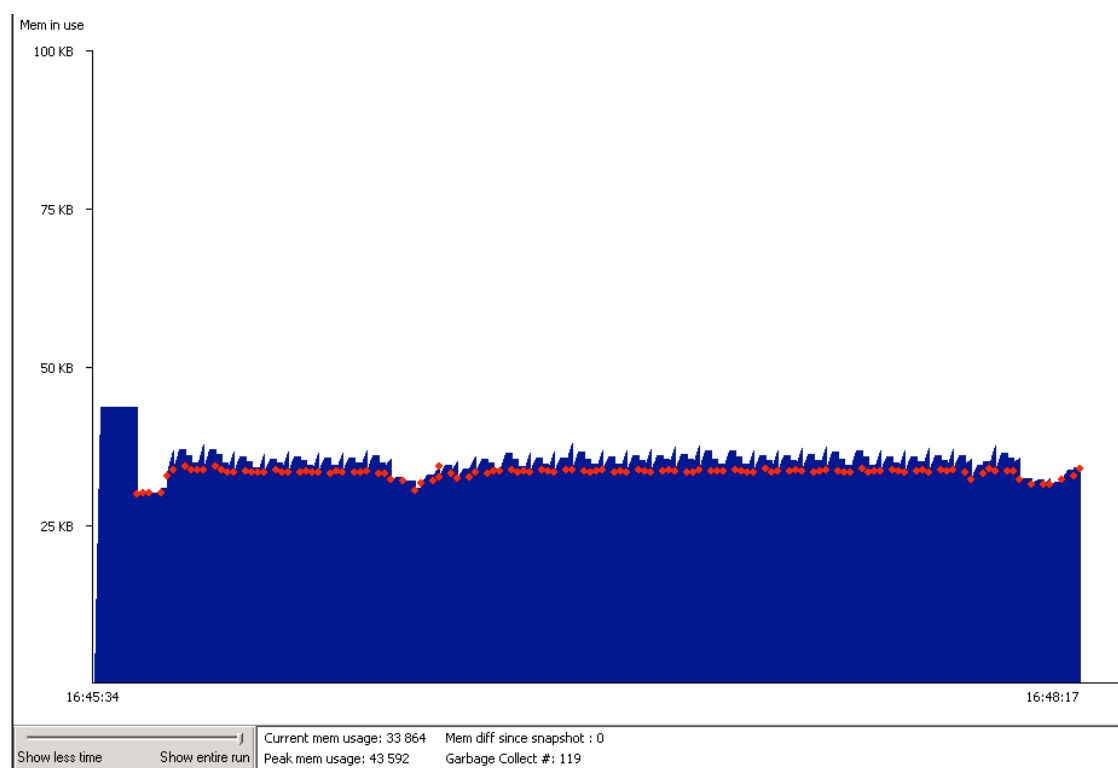
Zaměřili jsme se tedy spíše na hospodárné využití paměti programem. K testování jsme použili program Purify.

Method	Calls	Current method bytes allocated	Number of Objects	Class
DrawPanel.paintsmith	4 468	322 064	76	smitak04.DrawPanel
smitak04.DrawContr...	1	24 776	211	smitak04.DrawContr...
smitak04.DrawSwitc...	1	1 968	19	smitak04.DrawSwitc...
DrawTest.init	1	1 792	4	smitak04.DrawTest
DrawTest.main	1	680	5	smitak04.DrawTest
smitak04.DrawPanel	1	552	9	smitak04.DrawPanel
DrawSWRPanel.pain...	51	232	10	smitak04.DrawSWR...
DrawPanel.FormatN...	8 936	48	2	smitak04.DrawPanel
DrawControls.Format...	45 756	48	2	smitak04.DrawContr...
DrawControls.action...	5	24	0	smitak04.DrawContr...
DrawSWRPanel.upd...	51	24	0	smitak04.DrawSWR...
DrawControls.itemSt...	4	24	0	smitak04.DrawContr...
DrawPanel.mousePr...	3	24	0	smitak04.DrawPanel
DrawPanel.xy2gamm...	1 270	24	0	smitak04.DrawPanel
smitak04.DrawSWR...	1	24	0	smitak04.DrawSWR...
.Root.	0	0	0	(None)
.Thread-1.	0	0	0	(None)
.Thread-0.	0	0	0	(None)
.thread_3.	0	0	0	(None)
.DestroyJavaVM.	0	0	0	(None)
.AWT-EventQueue-0.	0	0	0	(None)
DrawPanel.paint	2	0	0	smitak04.DrawPanel
DrawSWRPanel.paint	2	0	0	smitak04.DrawSWR...
DrawPanel.update	4 468	0	0	smitak04.DrawPanel
DrawSwitches.itemSt...	6	0	0	smitak04.DrawSwitc...
DrawPanel.mouseDr...	1 259	0	0	smitak04.DrawPanel
DrawTest\$1.window...	1	0	0	smitak04.DrawTest\$1
DrawPanel.paintcoord	4 468	0	0	smitak04.DrawPanel
DrawPanel.radial	165 457	0	0	smitak04.DrawPanel
DrawPanel.gacirc	4 492	0	0	smitak04.DrawPanel
DrawPanel.xcirc	71 488	0	0	smitak04.DrawPanel
DrawPanel.rcirc	40 212	0	0	smitak04.DrawPanel
DrawPanel.bcirc	4 603	0	0	smitak04.DrawPanel
DrawPanel.gcirc	4 540	0	0	smitak04.DrawPanel
DrawControls.recalc...	1 271	0	0	smitak04.DrawContr...
DrawSWRPanel.Ne...	1 271	0	0	smitak04.DrawSWR...
DrawPanel.xyinside	1 273	0	0	smitak04.DrawPanel
DrawPanel.GammaC...	2	0	0	smitak04.DrawPanel
DrawPanel.yCoVis	2	0	0	smitak04.DrawPanel
DrawPanel.GammaC...	2	0	0	smitak04.DrawPanel
DrawPanel.NewGam...	7	0	0	smitak04.DrawPanel
DrawControls.recalc...	3	0	0	smitak04.DrawContr...

Výsledek Purify

Z výsledné zprávy je patrné, že nejvíce paměti spotřebuje funkce `DrawPanel.paintsmith`, což je funkce starající se o vykreslování Smithova diagramu.

Pro úplnost přikládáme ještě graf znázorňující spotřebu paměti v čase.



Průběžná spotřeba paměti

Použijeme-li informace získané z obou grafů, je vidět, že největší spotřebu má funkce `DrawPanel.paintsmith`. Ta se stará o vykreslení Smithova grafu.

Tato funkce je volána zbytečně často (celkem 4 468 krát). Pokaždé si alokuje množství proměnných (lze zjistit pohledem do zdrojového kódu), které jsou následně zahozeny a odstraněny GC. Z druhého grafu je patrné, že ten byl zavolán 119 během necelých 3 minut.

ZHODNOCENÍ

Jako problematická byla identifikována metoda `DrawPanel.paintsmith`, která je volána příliš často a zbytečně si alokuje mnoho lokálních proměnných, které jsou následně uvolněny.

Výkonnější kód

Již pohled na zatížení procesoru při jejím spuštění naznačuje, že aplikace, při své relativní jednoduchosti, není z hlediska výkonu příliš optimálně napsána.



Spotřeba CPU webového prohlížeče bez aplikace v %



Spotřeba CPU webového prohlížeče se spuštěnou aplikací v %

Pro přesnou a detailní analýzu jsme použili program Quantify.

Method	Calls	Method time	M+D time	M time (% of Focus)	M+D time (% of Focus)	Avg M time	Min M time	Max M time
DrawPanel.paintsmith	2 002	3294013,36	5570302,21	55,18	93,30	1645,36	0,30	332846,13
DrawPanel.rcirc	18 018	1424598,88	1510758,55	23,86	25,31	79,06	0,00	6481,81
JVM Garbage Collector	350	334286,94	2311170,47	5,60	38,71	955,11	605,19	19143,55
DrawPanel.xcirc	32 032	274316,31	353231,58	4,59	5,92	8,56	0,00	1240,98
DrawTest.main	1	173612,30	187206,49	2,91	3,14	173612,30	173612,30	173612,30
DrawPanel.gcirc	2 179	158314,32	171675,28	2,65	2,88	72,65	0,00	2037,39
DrawPanel.gcirc	2 002	156480,25	167270,02	2,62	2,80	78,16	0,00	2013,16
DrawSWRPanel.NewGamma	1 701	60456,20	63168,10	1,01	1,06	35,54	0,00	985,12
DrawSWRPanel.paintvoltage	115	35704,63	42186,60	0,60	0,71	310,47	0,00	1725,31
DrawControls.FormatNmb	61 236	21102,41	37760,02	0,35	0,63	0,34	0,00	2905,38
DrawPanel.bcirc	2 002	12269,82	14788,33	0,21	0,25	6,13	0,00	650,56
smitak04.DrawControls	1	9026,72	9026,72	0,15	0,15	9026,72	9026,72	9026,72
DrawPanel.update	2 002	6715,46	5747247,74	0,11	96,27	3,35	0,01	2238,82
DrawControls.recalculate	1 701	2480,56	103408,68	0,04	1,73	1,46	0,16	144,82
DrawPanel.FormatNmb	4 004	2007,07	4452,75	0,03	0,07	0,50	0,00	224,58
DrawTest.init	1	1900,24	10927,00	0,03	0,18	1900,24	1900,24	1900,24
DrawPanel.paintcoord	2 002	1081,02	1081,02	0,02	0,02	0,54	0,00	176,63
DrawPanel.radial	74 782	719,43	719,43	0,01	0,01	0,01	0,00	77,24
DrawSWRPanel.update	115	604,77	5747851,53	0,01	96,28	5,26	0,01	603,79
DrawPanel.mouseDragged	1 671	325,86	2614985,57	0,01	43,80	0,19	0,01	119,31
DrawPanel.xy2gammazyswr	1 701	18,48	27,69	0,00	0,00	0,01	0,01	3,87
DrawPanel.xyinside	1 723	9,30	9,30	0,00	0,00	0,01	0,00	1,90
DrawPanel.mousePressed	18	0,31	2633793,56	0,00	44,12	0,02	0,02	0,02
DrawControls.actionPerformed	6	0,08	5748510,77	0,00	96,29	0,01	0,01	0,01
DrawPanel.mouseClicked	4	0,07	1084128,61	0,00	18,16	0,02	0,02	0,02
DrawPanel.NewGamma	7	0,06	0,15	0,00	0,00	0,01	0,01	0,01
DrawSwitches.itemStateChan...	4	0,03	5336683,66	0,00	89,39	0,01	0,01	0,01
DrawControls.recalculate2	4	0,03	2924,49	0,00	0,05	0,01	0,01	0,01
DrawControls.itemStateChang...	2	0,02	2848370,87	0,00	47,71	0,01	0,01	0,01
DrawPanel.GammaCoVis	4	0,02	0,02	0,00	0,00	0,00	0,00	0,00
smitak04.DrawPanel	1	0,01	0,02	0,00	0,00	0,01	0,01	0,01
DrawPanel.paint	1	0,00	5748510,78	0,00	96,29	0,00	0,00	0,00
DrawSWRPanel.paint	1	0,00	5748510,77	0,00	96,29	0,00	0,00	0,00
DrawTest\$1.windowClosing	1	0,00	0,00	0,00	0,00	0,00	0,00	0,00
DrawPanel.register	1	0,00	0,00	0,00	0,00	0,00	0,00	0,00
smitak04.DrawSwitches	1	0,00	0,00	0,00	0,00	0,00	0,00	0,00
smitak04.DrawSWRPanel	1	0,00	0,00	0,00	0,00	0,00	0,00	0,00
smitak04.DrawTest	1	0,00	0,00	0,00	0,00	0,00	0,00	0,00
smitak04.DrawTest\$1	1	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Výsledek Quantify

Z provedené analýzy je patrné, že přes 80 % času tráví program v třídě DrawPanel, konkrétně v metodě paintsmith, starající se o vykreslení Smithova diagramu a jí pomocných metodách.

ZHODNOCENÍ

Obdobně jako v předchozí analýze se ukázala jako problematická metoda `DrawPanel.paintsmith`. Jedná se o poměrně náročnou metodu, starající se o vykreslení Smithova diagramu. Vzhledem ke své náročnosti byla ale volána asi 2000x během krátkého běhu programu.

Výsledkem této a předchozí analýzy je tedy doporučení přepsat inkriminovanou metodu tak, aby byla volána méně často, typicky pouze při nutnosti změny zobrazení, a aby lépe hospodařila s pamětí, například globálním vytvoření proměnných a pouhou jejich změnou oproti neustálé alokaci a uvolňování nové paměti, což také snižuje výkon voláním GC.

Nevolané části programu

Ke zjištění nevolaných částí jsme použili program PureCovarege. Během testu jsme se pokusili využít všech možností programu a najít tak jeho zbytečné a zapomenuté metody.

Coverage Item	Calls	Methods Missed	Methods Hit	% Methods Hit
Run @ 09.04.2008 17:16:28 -jar JavaSource.jar	510111	14	39	73,58
JVM internals	939	0	1	100,00
(Unknown Directory)	939	0	1	100,00
(Unknown File)	939	0	1	100,00
JVM Garbage Collector	939		hit	
smitak04.DrawControls	40888	0	7	100,00
(Unknown Directory)	40888	0	7	100,00
DrawTest.java	40888	0	7	100,00
smitak04.DrawControls.FormatNmb(double)	39722		hit	
smitak04.DrawControls.actionPerformed(java.awt.event.ActionEvent)	34		hit	
smitak04.DrawControls.itemStateChanged(java.awt.event.ItemEvent)	7		hit	
smitak04.DrawControls.recalculate(double, double)	1103		hit	
smitak04.DrawControls.recalculate2(double, double)	19		hit	
smitak04.DrawControls.recalculatef(double)	2		hit	
smitak04.DrawControls.smitak04.DrawControls(smitak04.DrawPanel, smi tak04.DrawSWRPanel)	1		hit	
smitak04.DrawPanel	467012	11	19	63,33
(Unknown Directory)	467012	11	19	63,33
DrawTest.java	467012	11	19	63,33
smitak04.DrawPanel.FormatNmb(double)	13238		hit	
smitak04.DrawPanel.GammaCoVis(boolean)	2		hit	
smitak04.DrawPanel.GammaCuVis(boolean)	0	missed		
smitak04.DrawPanel.NewGamma(double, double)	31		hit	
smitak04.DrawPanel.bcirc(java.awt.Graphics, double)	6619		hit	
smitak04.DrawPanel.gacirc(java.awt.Graphics, double)	6691		hit	
smitak04.DrawPanel.gcirc(java.awt.Graphics, double)	6619		hit	
smitak04.DrawPanel.mouseClicked(java.awt.event.MouseEvent)	0	missed		
smitak04.DrawPanel.mouseDragged(java.awt.event.MouseEvent)	1062		hit	
smitak04.DrawPanel.mouseEntered(java.awt.event.MouseEvent)	0	missed		
smitak04.DrawPanel.mouseExited(java.awt.event.MouseEvent)	0	missed		
smitak04.DrawPanel.mouseMoved(java.awt.event.MouseEvent)	0	missed		
smitak04.DrawPanel.mousePressed(java.awt.event.MouseEvent)	7		hit	
smitak04.DrawPanel.mouseReleased(java.awt.event.MouseEvent)	0	missed		
smitak04.DrawPanel.paint(java.awt.Graphics)	11		hit	
smitak04.DrawPanel.paintcoord(java.awt.Graphics, java.lang.String)	6619		hit	
smitak04.DrawPanel.paintsmith(java.awt.Graphics)	6619		hit	
smitak04.DrawPanel.radial(java.awt.Graphics, double, int)	245191		hit	
smitak04.DrawPanel.rcirc(java.awt.Graphics, double)	59571		hit	
smitak04.DrawPanel.register(smitak04.DrawControls)	1		hit	
smitak04.DrawPanel.smitak04.DrawPanel()	1		hit	
smitak04.DrawPanel.update(java.awt.Graphics)	6619		hit	
smitak04.DrawPanel.xcirc(java.awt.Graphics, double)	105904		hit	
smitak04.DrawPanel.xy2gammazyswr(int, int, int, int)	1100		hit	
smitak04.DrawPanel.xyactrecalculate()	0	missed		
smitak04.DrawPanel.xyinside(int, int)	1107		hit	
smitak04.DrawPanel.yCoVis(boolean)	0	missed		
smitak04.DrawPanel.yCuVis(boolean)	0	missed		
smitak04.DrawPanel.zCoVis(boolean)	0	missed		
smitak04.DrawPanel.zCuVis(boolean)	0	missed		
smitak04.DrawSWRPanel	1264	0	5	100,00
(Unknown Directory)	1264	0	5	100,00
DrawTest.java	1264	0	5	100,00
smitak04.DrawSWRPanel.NewGamma(double, double)	1103		hit	
smitak04.DrawSWRPanel.paint(java.awt.Graphics)	10		hit	
smitak04.DrawSWRPanel.paintvoltage(java.awt.Graphics)	75		hit	
smitak04.DrawSWRPanel.smitak04.DrawSWRPanel()	1		hit	
smitak04.DrawSWRPanel.update(java.awt.Graphics)	75		hit	
smitak04.DrawSwitches	3	1	2	66,67
(Unknown Directory)	3	1	2	66,67
DrawTest.java	3	1	2	66,67
smitak04.DrawSwitches.itemStateChanged(java.awt.event.ItemEvent)	2		hit	
smitak04.DrawSwitches.paint(java.awt.Graphics)	0	missed		
smitak04.DrawSwitches.smitak04.DrawSwitches(smitak04.DrawPanel)	1		hit	
smitak04.DrawTest	3	2	3	60,00
(Unknown Directory)	3	2	3	60,00
DrawTest.java	3	2	3	60,00
smitak04.DrawTest.destroy()	0	missed		
smitak04.DrawTest.getAppletInfo()	0	missed		
smitak04.DrawTest.init()	1		hit	
smitak04.DrawTest.main(java.lang.String [])	1		hit	
smitak04.DrawTest.smitak04.DrawTest()	1		hit	
smitak04.DrawTest\$1	2	0	2	100,00
(Unknown Directory)	2	0	2	100,00
DrawTest.java	2	0	2	100,00
smitak04.DrawTest\$1.smitak04.DrawTest\$1()	1		hit	
smitak04.DrawTest\$1.windowClosing(java.awt.event.WindowEvent)	1		hit	

Výsledek programu PureCoverage

Z testu je vidět, že některé třídy nebyly volány. Ty lze rozdělit do 3 skupin.

Po vyhodnocení bylo zjištěno, že přes veškerou snahu nebyly některé funkce využity. Ty patří do první skupiny. To jsou funkce, které jsou v kódu použity, ale nebyly pokryty testem.























Druhá skupina jsou funkce, které program nepoužívá, ale v rámci dědění je musí implementovat.

Poslední skupina je specifická pro způsob běhu – applet/aplikace.

Manuální testování

Manuální testování jsme použili pro základní kontrolu funkčnosti programu, jako je spustitelnost formou appletu a fungování grafického rozhraní.

Pro testování jsme použili program Rational ManualTest od firmy IBM. V něm jsme vytvořili testovací scénář, který byl následně proveden.

	Type	Note	Description
1			Spustíte webový prohlížeč Internet Explorer verze 7
2			Napište adresu www.uzdroje.ic.cz
3			Zobrazen neinicializovaný vzhled programu. (1)
4			Zobrazen neinicializovaný vzhled programu. (2)
5			Zobrazen neinicializovaný vzhled programu. (3)
6			Zobrazen neinicializovaný vzhled programu. (4)
7			Stiskněte ENTER.
8			Program inicializován. (1)
9			Program inicializován. (2)
10			Program inicializován. (3)
11			Program inicializován. (4)
12			Klikněte kamkoliv v diagramu.
13			Správnost funkce diagramu. (1)
14			Správnost funkce diagramu. (2)
15			Správnost funkce diagramu. (3)
16			Vyzkoušejte všechny kombinace zaškrtnutí checkboxů v levé spodní části programu.
17			Správnost funkce checkboxů.
18			Pokuste se změnit hodnoty kapacit a indukčností.
19			Kapacita a indukčnost nejsou editovatelné.
20			Zkontrolujte, že správně fungují pulldowny pro min a max, a různé formáty vlnové délky.
21			Správná funkčnost pulldownů.
22			Zavřete okno prohlížeče.

Testovací scénář (levé část)

Expected Result
Spuštěn zadaný prohlížeč.
Otevře se stránka s Java appletem Smith chart Java applet.
Zobrazil se panel s hodnotami, ve kterém byla předvyplněná hodnota $Z_0=50$.
Pod tímto panelem je panel pro zobrazení PSW.
Vlevo dole je panel s checkboxy pomocí kterých se dá určit, které osy se mají ve Smithově diagramu zobrazit.
Defaultně jsou zaškrtnuty: Cursors>Gamma,z,y; Grids>z.
V pravé části appletu se zobrazí Smithův diagram.
V pravé části appletu je zobrazen Smithův diagram.
Kurzor je umístěn uprostřed diagramu.
Jsou zobrazeny ty osy, které jsou zaškrtnuty v panelu s checkboxy.
V panelu s hodnotami se objevily hodnoty platné pro přizpůsobený obvod.
Přepočítají se hodnoty v panelu hodnot. Změní se popisek u vybraného bodu. Překreslí se panel PSW.
Přepočítaly se hodnoty v panelu hodnot.
Změnil se popisek u vybraného bodu.
Překreslil se panel PSW.
Zobrazí se vždy správné osy a popisky bodu
Zobrazily se vždy správné osy a popisky bodu v grafu.
Tato pole nejsou editovatelná.
Nepodařilo se zadat hodnotu.
Změní se příslušné hodnoty.
Změnily se příslušné hodnoty.
Okno se zavře.

Testovací scénář (pravá část)

Event Type	Result
Computer Start	Fail
Script Start (ManualniTest_v2)	Fail
Manual Step (Spustíte webový prohlížeč Internet Explorer verze 7)	Completed
Manual Step (Napište adresu www.uzdroje.ic.cz)	Completed
Verification Point (Zobrazen neinicializovaný vzhled programu. (1) - Manual)	Fail
Verification Point (Zobrazen neinicializovaný vzhled programu. (2) - Manual)	Pass
Verification Point (Zobrazen neinicializovaný vzhled programu. (3) - Manual)	Pass
Verification Point (Zobrazen neinicializovaný vzhled programu. (4) - Manual)	Pass
Manual Step (Stiskněte ENTER.)	Completed
Verification Point (Program inicializován. (1) - Manual)	Fail
Verification Point (Program inicializován. (2) - Manual)	Fail
Verification Point (Program inicializován. (3) - Manual)	Fail
Verification Point (Program inicializován. (4) - Manual)	Pass
Manual Step (Klikněte kamkoliv v diagramu.)	Completed
Verification Point (Správnost funkce diagramu. (1) - Manual)	Pass
Verification Point (Správnost funkce diagramu. (2) - Manual)	Pass
Verification Point (Správnost funkce diagramu. (3) - Manual)	Pass
Manual Step (Vyzkoušejte všechny kombinace zaškrtnutí checkboxů v levé spod...	Completed
Verification Point (Správnost funkce checkboxů. - Manual)	Pass
Manual Step (Pokuste se změnit hodnoty kapacit a indukčností.)	Completed
Verification Point (Kapacita a indukčnost nejsou editovatelné. - Manual)	Fail
Manual Step (Zkontrolujte, že správně fungují pulldowny pro min a max, a různ)	Completed
Verification Point (Správná funkčnost pulldownů. - Manual)	Pass
Manual Step (Zavřete okno prohlížeče.)	Completed
Script End (ManualniTest_v2)	Fail
Computer End	Fail

Výsledek testu

ZHODNOCENÍ

Manuálním testem jsme potvrdili funkčnost programu ve formě JAVA appletu.

Během testu jsme ovšem odhalili několik chyb:

- po spuštění není předvyplněna hodnota Z0
- po aktivaci programu není správně inicializováno zobrazení grafu
- do polí, které slouží pouze pro zobrazení výsledku, lze zapisovat

Testování grafického prostředí

Za pomoci testovacího nástroje Rational Robot jsme otestovali správnou funkčnost programu.

Provedené testy lze rozdělit do dvou skupin. První typ testoval reakci programu na nestandardní vstup. Druhý potom správnost výpočtu pomocí externího orákula (zde matlabovský skript).

REAKCE NA NESTANDARDNÍ VSTUP

Jelikož program provádí praktický matematický výpočet, je rozsah zadávaných hodnot omezen. V případě, že uživatel zadá chybnou hodnotu, měl by ji program ignorovat. Hodnota by neměla být akceptována. Skript tedy testuje, zda po zadání špatné hodnoty, je chybná hodnota přepsána hodnotou původní.

První test kontroluje, zda je ignorován znakový vstup.

Principiální ukázka:

hodnota="ABCD"

EditBox DbClick, "ObjectIndex=1", "Coords=19,8"

InputKeys hodnota+"{ENTER}"

Result = EditBoxVP (CompareData, "ObjectIndex=1", "VP=Object Data; Wait=2,2; \ ExpectedResult=FAIL")

Event Type	Result
Computer Start	Warning
Script Start (Extremni hodnoty_v2)	Pass
Application Start	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Playback Warning	Warning
Verification Point (- Object Data)	Pass
Playback Warning	Warning
Verification Point (- Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Script End (Extremni hodnoty_v2)	Pass
Computer End	Warning

Výsledek - znakový vstup

Test proběhl v pořádku. Program správně ignoruje textové hodnoty.

Druhý test kontroluje, zda program ignoruje záporné hodnoty u vybraných parametrů.

Principiální ukázka:

hodnota="-0.9"

EditText DbClick, "ObjectIndex=1", "Coords=19,8"

InputKeys hodnota+"{ENTER}"

Result = EditTextVP (CompareNumeric, "ObjectIndex=1", "VP=Zo; Wait=2,2;\nValue="+hodnota+"; ExpectedResult=FAIL")

Event Type	Result
Computer Start	Fail
Script Start (Extremni hodnoty_v3)	Fail
Application Start	Pass
Verification Point (Zo - Alphanumeric)	Fail
Verification Point (Yo - Alphanumeric)	Pass
Verification Point (Re(z) - Alphanumeric)	Pass
Verification Point (Izl - Alphanumeric)	Pass
Verification Point (Angle_deg - Alphanumeric)	Fail
Verification Point (Re(z)_Ohm - Alphanumeric)	Fail
Verification Point (Izl_Ohm - Alphanumeric)	Pass
Verification Point (Angle_rad - Alphanumeric)	Fail
Verification Point (Iyl - Alphanumeric)	Pass
Verification Point (Angle_deg - Alphanumeric)	Fail
Verification Point (Re(y)_Ohm - Alphanumeric)	Fail
Verification Point (Iyl_mS - Alphanumeric)	Pass
Verification Point (Angle_rad - Alphanumeric)	Pass
Verification Point (IGI - Alphanumeric)	Pass
Verification Point (Angle_rad - Alphanumeric)	Fail
Verification Point (Angle_deg - Alphanumeric)	Fail
Verification Point (VSWR - Alphanumeric)	Pass
Verification Point (Emax - Alphanumeric)	Pass
Verification Point (Emin - Alphanumeric)	Pass
Verification Point (Freq - Alphanumeric)	Fail
Verification Point (Lambda - Alphanumeric)	Pass
Script End (Extremni hodnoty_v3)	Fail
Computer End	Fail

Výsledek - záporný vstup

Jak je vidět z výsledku testu, některé parametry neignorovaly záporné hodnoty a akceptovaly je.

Poslední test kontroloval zaokrouhlování vysokých hodnot.

Principiální ukázka:

hodnota="10000"

EditBox DbClick, "ObjectIndex=1", "Coords=19,8"

InputKeys hodnota+"{ENTER}"

Result = EditBoxVP (CompareData, "ObjectIndex=1", "VP=Object Data; Wait=2,2; \ ExpectedResult=FAIL")

'Object Data=">1000"

Event Type	Result
Computer Start	Fail
Script Start (Extremni hodnoty_v4)	Fail
Application Start	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Fail
Playback Warning	Warning
Verification Point (- Object Data)	Fail
Playback Warning	Warning
Verification Point (- Object Data)	Fail
Verification Point (Object Data - Object Data)	Fail
Verification Point (Object Data - Object Data)	Fail
Verification Point (Object Data - Object Data)	Fail
Verification Point (Object Data - Object Data)	Fail
Verification Point (Object Data - Object Data)	Fail
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Fail
Verification Point (Object Data - Object Data)	Fail
Verification Point (Object Data - Object Data)	Fail
Verification Point (Object Data - Object Data)	Fail
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Verification Point (Object Data - Object Data)	Pass
Script End (Extremni hodnoty_v4)	Fail
Computer End	Fail

Výsledek - vysoké hodnoty

Z výsledku je opět patrné, že program nezafungoval dle očekávání.

SPRÁVNOST VÝPOČTU

První skript kontroluje správnost vnitřního přepočtu hodnot. Vychází ze znalosti vnitřního fungování programu, který ukládá pouze vybrané parametry a ostatní z nich dopočítává. V našem případě je test koncipován tak, že je zadán parametr a je zkontrolováno, zda po provedení vnitřních přepočtů zůstala správná hodnota.

Principiální ukázka:

TextBox DbClick, "ObjectIndex=1", "Coords=19,8"

InputKeys "23{ENTER}"

Result = TextBoxVP (CompareNumeric, "ObjectIndex=1", "VP=Zo;Wait=2,10;Value=23")

Event Type	Result
Computer Start	Fail
Script Start (SpravnyPrepocet)	Fail
Application Start	Pass
Verification Point (Zo - Alphanumeric)	Pass
Verification Point (Yo - Alphanumeric)	Fail
Verification Point (Re(z) - Alphanumeric)	Pass
Verification Point (Im(z) - Alphanumeric)	Pass
Verification Point (Izl - Alphanumeric)	Pass
Verification Point (Angle_deg - Alphanumeric)	Pass
Verification Point (Re(z)_Ohm - Alphanumeric)	Pass
Verification Point (Im(z)_Ohm - Alphanumeric)	Pass
Verification Point (Izl_Ohm - Alphanumeric)	Fail
Verification Point (Angle_rad - Alphanumeric)	Pass
Verification Point (Im(y) - Alphanumeric)	Pass
Verification Point (Iyl - Alphanumeric)	Pass
Verification Point (Angle_deg - Alphanumeric)	Pass
Verification Point (Re(y)_Ohm - Alphanumeric)	Pass
Verification Point (Im(y)_Ohm - Alphanumeric)	Pass
Verification Point (Iyl_mS - Alphanumeric)	Pass
Verification Point (Angle_rad - Alphanumeric)	Fail
Verification Point (Re(g) - Alphanumeric)	Pass
Verification Point (Im(G) - Alphanumeric)	Pass
Verification Point (IGl - Alphanumeric)	Pass
Verification Point (Angle_rad - Alphanumeric)	Pass
Verification Point (IGl_dB - Alphanumeric)	Pass
Verification Point (Angle_deg - Alphanumeric)	Pass
Verification Point (VSWR - Alphanumeric)	Pass
Verification Point (VSWR_dB - Alphanumeric)	Pass
Verification Point (Emax - Alphanumeric)	Pass
Verification Point (Emin - Alphanumeric)	Pass
Verification Point (Freq - Alphanumeric)	Pass
Verification Point (Lambda - Alphanumeric)	Fail
Script End (SpravnyPrepocet)	Fail
Computer End	Fail

Výsledek - vnitřní přepočet

Byla zjištěna chybná funkce programu u několika parametrů.

Druhý skript kontroluje vypočtené hodnoty pomocí externího zdroje dat. Skript zadá Z_0 , f , $\text{Re}\{z\}$, $\text{Im}\{z\}$ a porovná výsledky.

Event Type	Result
Computer Start	Fail
Script Start (TestOrakulum)	Fail
Application Start	Pass
Verification Point (Yo - Alphanumeric)	Pass
Verification Point (Izl - Alphanumeric)	Pass
Verification Point (Angle_deg - Alphanumeric)	Pass
Verification Point (Re(z)_Ohm - Alphanumeric)	Pass
Verification Point (Im(z)_Ohm - Alphanumeric)	Pass
Verification Point (Izl_Ohm - Alphanumeric)	Pass
Verification Point (Angle_rad - Alphanumeric)	Pass
Verification Point (Im(y) - Alphanumeric)	Fail
Verification Point (Iyl - Alphanumeric)	Fail
Verification Point (Angle_deg - Alphanumeric)	Fail
Verification Point (Re(y)_Ohm - Alphanumeric)	Fail
Verification Point (Im(y)_Ohm - Alphanumeric)	Fail
Verification Point (Iyl_mS - Alphanumeric)	Fail
Verification Point (Angle_rad - Alphanumeric)	Fail
Verification Point (Re(g) - Alphanumeric)	Pass
Verification Point (Im(G) - Alphanumeric)	Pass
Verification Point (IGl - Alphanumeric)	Pass
Verification Point (Angle_rad - Alphanumeric)	Pass
Verification Point (IGl_dB - Alphanumeric)	Fail
Verification Point (Angle_deg - Alphanumeric)	Pass
Verification Point (VSWR - Alphanumeric)	Pass
Verification Point (VSWR_dB - Alphanumeric)	Pass
Verification Point (Emax - Alphanumeric)	Pass
Verification Point (Emin - Alphanumeric)	Pass
Verification Point (Lambda - Alphanumeric)	Fail
Script End (TestOrakulum)	Fail
Computer End	Fail

Výsledek - orákulum

Vypočtené hodnoty se neshodují s porovnávanými.

ZHODNOCENÍ

Pomocí testů grafického rozhraní jsme odhalili řadu chyb:

- není ošetřeno zadání hodnot mimo stanovený rozsah
 - záporné hodnoty
 - vysoká čísla
- vnitřní přepoččet hodnot obsahuje chyby
- algoritmus výpočtu hodnot obsahuje chyby

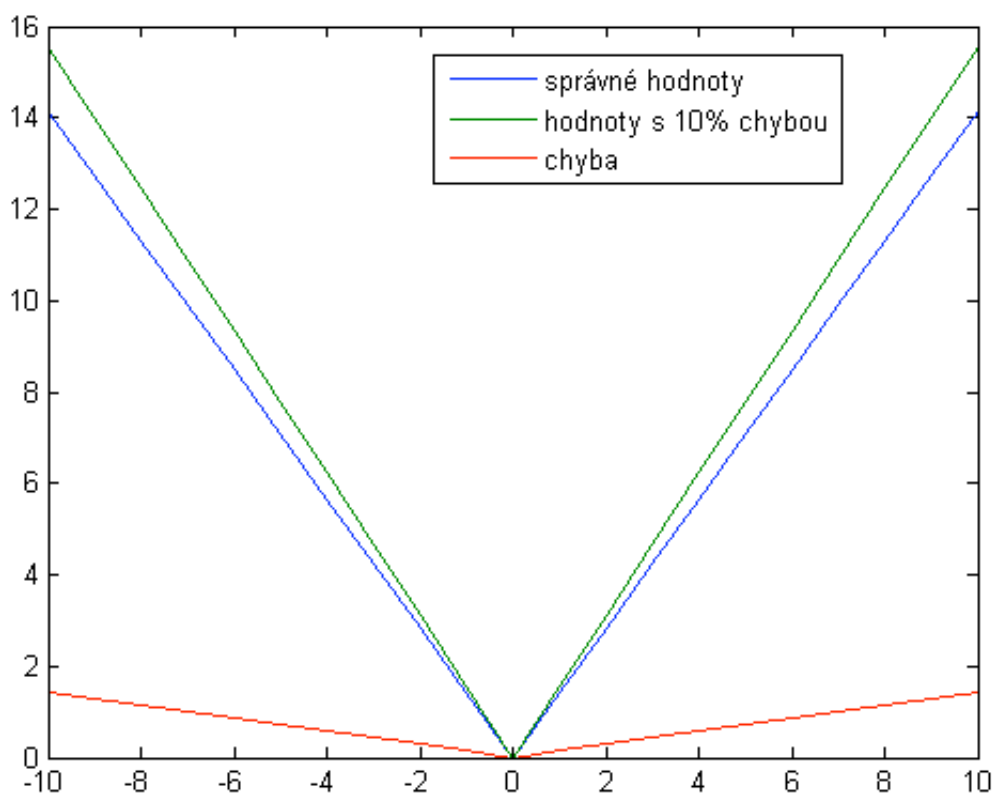
Šíření chyb

V tomto programu vznikají hlavně zaokrouhlovací chyby a chyby při přepočtech.

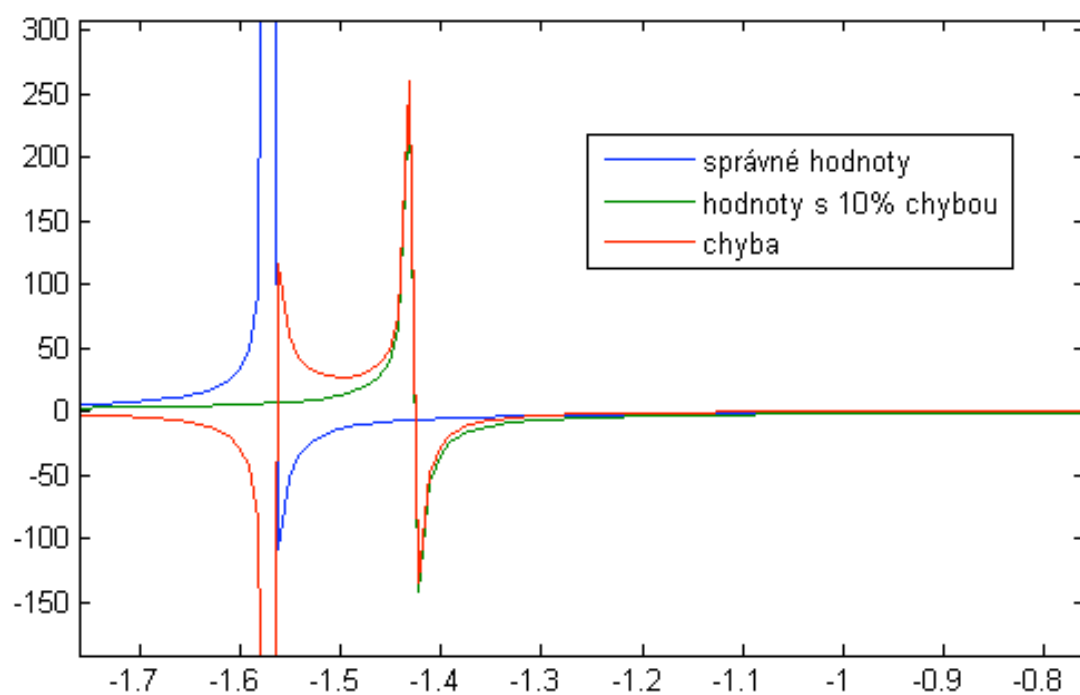
Kterýkoliv vstupní parametr je přepočítávána na dvojici argumentů GMAG a GPHA (absolutní hodnota a fáze parametru G. Z těch se potom zpětně dopočítávají všechny ostatní hodnoty. Pro vykreslení v diagramu je potřeba dvojice parametrů GR a GI – to znamená reálnou a imaginární složku parametru G. Chyba se tak projevuje dvojnásobně – poprvé při přepočtu na GMAG a GPHA a potom při přepočtu na ostatní parametry.

Při těchto přepočtech jsou používány základní aritmetické operace a goniometrické funkce.

Největší chyba v panelu hodnot je dána zaokrouhlením – je použito zobrazení min 3 nenulových hodnot na max 3 desetinná místa. Zajímat nás tedy bude především přepočet při zadávání hodnot pomocí Smithova diagramu – přepočítáváme reálnou a imaginární složku na abs. hodnotu a fázi.



Šíření chyby při výpočtu absolutní hodnoty



Šíření chyby při výpočtu *tangens*

Závěr

Provedli jsme testování vybraného programu. Na začátku testování nebyly známy u daného programu žádné funkční chyby.

Během testování se projevily výkonnostní problémy. Pomocí programů Purify a Quatify jsme odhalili problém v nevhodně navrženém vykreslování Smithova diagramu, které je spouštěno zbytečně často. Při svém spuštění též stále znova alokuje paměť, aby ji vzápětí uvolnil.

Z funkčního hlediska jsme, pomocí programů ManualTester a Robot, oproti specifikaci odhalili odlišnost fungování po inicializaci programu. Dále jsme objevili chyby v ošetření zadávaných údajů a chyby při výpočtech.

Získané informace poslouží autorce programu k jeho zlepšení.