

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

LATVIEŠU VALODAS VĀRDU LOCĪTĀJS

KVALIFIKĀCIJAS DARBS

Autors:
Pēteris Ņikiforovs
Stud. apl. nr. pc09002

Vadītājs:
Normunds Grūzītis
Dr.sc.comp.

RĪGA 2011

Anotācija

Darbs „Latviešu valodas vārdu locītājs” apraksta programmatūras bibliotēkas izstrādi, kas ļauj izlocīt latviešu valodas lietvārdus, īpašības vārdus, darbības vārdus un divdabjus, zinot to pamatformas un morfoloģiskās īpašības.

Atslēgvārdi: latviešu valoda, morfoloģija

Abstract

The paper „Latvian word inflector” describes the development of a software library for inflecting nouns, adjectives, verbs and participles in Latvian.

Keywords: Latvian, morphology, inflector

Saturs

Ievads.....	4
1. Programmatūras prasību specifikācija.....	5
1.1. Ievads	5
1.1.1. Nolūks	5
1.1.2. Darbības sfēra.....	5
1.1.3. Definīcijas, saīsinājumi un apzīmējumi	5
1.1.4. Saistība ar citiem dokumentiem	7
1.2. Vispārējais apraksts	8
1.2.1. Produkta perspektīva.....	8
1.2.2. Produkta funkcijas.....	8
1.2.3. Lietotāja raksturiesīmes	8
1.2.4. Pieņēmumi un atkarības	8
1.3. Funkcionālās prasības	8
1.3.1. Lietvārdi	9
1.3.2. Īpašības vārdi	15
1.3.3. Darbības vārdi	18
1.3.4. Divdabji.....	24
1.4. Nefunkcionālās prasības	29
1.4.1. Veiktspējas prasības	29
1.4.2. Ārējās saskarnes prasības	29
2. Programmatūras projektējuma apraksts.....	30
2.1. Ievads	30
2.1.1. Dokumenta nolūks	30
2.1.2. Darbības sfēra.....	30
2.1.3. Definīcijas, saīsinājumi un apzīmējumi	30
2.2. Saistība ar citiem dokumentiem.....	30

2.3. Dekompozīcijas apraksts	31
2.3.1. Ievads	31
2.3.2. Entītiņu dekompozīcija	32
2.4. Atkarību projektējums	34
2.5. Detalizētais projektējums.....	34
2.5.1. Vārdi.....	34
2.5.2. Lietvārdi	37
2.5.3. Īpašības vārdi	39
2.5.4. Darbības vārdi	42
2.5.5. Divdabji.....	43
3. Testēšanas dokumentācija	44
3.1. Īpašības vārdu testi.....	44
3.2. Lietvārdu testi	45
3.3. Darbības vārdu testi	45
3.4. Divdabju testi.....	47
4. Programmatūras pirmkoda fragmenti.....	48
5. Projekta organizācija	53
6. Konfigurācijas pārvaldība.....	54
7. Kvalitātes nodrošināšana	55
8. Darbietilpības novērtējums.....	56
9. Rezultāti.....	57
10. Secinājumi	58
Pateicības.....	59
Izmantotā literatūra un avoti.....	60

Ievads

Šī darba mērķis ir apkopot un aprakstīt programmatūras prasību specifikācijā latviešu valodas vārdu locīšanas likumus un, balstoties uz savāktajām prasībām, projektēt un praktiski izveidot vārdu locīšanas bibliotēku.

Izveidotajā programmatūras bibliotēkā tiek locītas tikai pamatvārdšķiras: lietvārdi, īpašības vārdi, darbības vārdi (un arī divdabji). Pārējās vārdšķiras ir vai nu nelokāmas, lokās paredzami un īpatnēji vai var izlocīt kā lietvārdus un īpašības vārdus.

Lai varētu izlocīt vārdus, bibliotēkai ir jāzina vārda pamatforma un morfoloģiskās īpašības, pēc kurām nosaka, kādu locīšanas paraugu izmantot. Šādu informāciju var atrast specializētās vārdnīcās.

Bibliotēkai ir vairāki praktiski pielietojumi:

- meklēšana visos vārda locījumos,
- vārda pārvēršana no vienskaitļa daudzskaitlī,
- automatizēta daudzu personvārdu locīšana,
- valodas apmācība,
- valodas spēles,
- vārdu analīze.

Kvalifikācijas darbā latviešu valodas vārdu locīšanas bibliotēka tika izveidota programmēšanas valodā C#, un visi izmantotie locīšanas likumi ir aprakstīti tabulu veidā programmatūras prasību specifikācijā un iespējamais projektējums objektorientētā programmēšanas valodā ir aprakstīts programmatūras projektējuma aprakstā. Izmantojot abus dokumentus, ir iespējams izveidot tādu pašu bibliotēku jebkurā citā programmēšanas valodā, kas atbalsta darbības ar teksta virknēm.

Darbs ir izstrādāts patstāvīgi un neatkarīgi no citiem līdzīgiem projektiem.

1. Programmatūras prasību specifikācija

1.1. Ievads

1.1.1. Nolūks

Programmatūras prasību specifikācija ir paredzēta izstrādājamās latviešu valodas vārdu locīšanas bibliotēkas prasību aprakstīšanai. Šis dokuments paredzēts programmatūras moduļa projektētājam, lai varētu izveidot projektējumu un valodniekam, lai saskaņotu gramatikas likumus ar funkcionalitāti, kā arī testētājiem.

1.1.2. Darbības sfēra

Šajā dokumentā definētas prasības latviešu valodas vārdu locīšanas bibliotēkai.

Programmatūras produkta mērķis ir nodrošināt bibliotēku, kas spētu automatizēti, zinot vārda patmatformu un morfoloģiskās īpašības, izlocīt vārdu (atgriežot visas iespējamās vārdformas) atbilstoši latviešu valodas gramatikas likumiem. Produkta paredzētais pielietojums ir pirmkoda bibliotēka, ko iekļauj, izmanto un iespējams papildina citi produkti.

1.1.3. Definīcijas, saīsinājumi un apzīmējumi

1.1.3.1. Definīcijas

1. Morfoloģija ir gramatikas daļa, kura aplūko vārda formu veidošanu un to paradigmas (piem., suns, suņi, sunim).
2. Morfēma ir morfoloģiski mazākā, nozīmīgākā valodas sistēmas vienība, vārda vai vārdformas daļa.
3. Sakne ir morfēma, kas kopīga veselai radniecīgu vārdu saimei. Parasti tā lokot nemainās. Sakne ir obligātā vārda sastāvdaļa.
4. Celms ir viss vārds bez galotnes.
5. Galotne ir morfēma, kas atrodas vārda beigās un kas, vārdu lokot, mainās.
6. Piedēklis ir morfēma starp sakni un galotni.
7. Izskaņa ir vārda daļa, kurā ietilpst vārda pēdējais piedēklis kopā ar galotni.
8. Pamatforma ir vārda forma, pēc kuras var noteikt gramatiskās īpašības un izlocīt vārdus.
9. Dzimte ir gramatiskā dzimte.

10. Skaitlis ir gramatiskais skaitlis, kas apzīmē daudzumu.
11. Locījums konkretizē vārda saistījumu ar kādu citu vārdu teikumā un atbild uz noteiktu jautājumu.
12. Lietvārds ir patstāvīgs vārds, kas priekšmetiskā veidā nosauc dažādas īstenības parādības. Priekšmetiskums realizējas tikai daļēji, jo ir tīri gramatisks.
13. Nelokāms lietvārds jeb deklinējams lietvārds ir tāds lietvārds, kurš tiek locīts pēc deklinācijas parauga.
14. Atgriezenisks lietvārds ir no atgriezeniska darbības vārda darināts lietvārds.
15. Personvārds ir īpašvārds (ar lielo sākumburtu rakstāms vārds), kas apzīmē cilvēka vārdu vai uzvārdu.
16. Deklinācija ir lietvārdu locījumu galotņu iedalījums, ņemot vērā dzimti un pamatformu galotnes.
17. Līdzskaņu mija ir vārda celma beigu daļas maiņa locījumos labskanības labad.
18. Īpašības vārds ir patstāvīgs, lokāms vārds, kas nosauc priekšmetu vai dzīvu būtni pazīmes.
19. Īpašības vārda noteiktā un nenoteiktā galotne norāda priekšmeta pazīmi kā zināmu vai nezināmu.
20. Īpašības vārda salīdzināmā pakāpe norāda, ka īpašība priekšmetam vai parādībai piemīt lielākā vai mazākā mērā nekā kādam citam līdzīgam priekšmetam vai parādībai.
21. Darbības vārds ir patstāvīgs vārds, kas izsaka darbību, stāvokli vai procesu.
22. Darbības vārda konjugācija ir darbības vārda formu maiņas modelis.
23. Darbības vārda grupa ir darbības vārdu sīkāks iedalījums.
24. Darbības vārda nenoteiksme nosauc darbību, bet nenorāda uz laiku, personu, skaitli.
25. Darbības vārda persona ir gramatikas jēdziens, kas atspoguļo runātāja attieksmes pret darbības veicēju. Ir trīs personas formas (pirmā, otrā, trešā) un divi skaitļi (vienskaitlis, daudzskaitlis).
26. Darbības vārda izteiksme atspoguļo darbības attiecinājumu pret īstenību.
27. Darbības vārda laiks ir gramatikas jēdziens, kas atspoguļo runātāja attieksmi pret runas momentu un laikposmiem pirms un pēc tā.
28. Darbības vārda kārtā izsaka sakarus starp darītāju, darbību un darbības objektu (priekšmetu).
29. Divdabis ir darbības vārda bezpersonas forma, kam ir darbības vārda un īpašības vārda vai apstākļa vārda pazīmes.
30. Vārdu locīšana ir visu iespējamo vārdformu izveidošana.

1.1.3.2. Saīsinājumi

- vīr.dz. – vīriešu dzimte
- siev.dz. – sieviešu dzimte
- dekl. – deklinācija
- vsk. – vienskaitlis
- dsk. – daudzskaitlis
- lietv. – lietvārds
- atgr. lietv. – atgriezenisks lietvārds
- nenot.g. – nenoteiktā galotne
- not.g. – noteiktā galotne

1.1.3.3. Apzīmējumi

- -s vai -us (domu zīme ar vienu vai diviem burtiem) apzīmē vārda galotni, domu zīme nav jāņem vērā.
- -iet vai -šana (domu zīme ar trim vai vairāk burtiem) apzīmē vārda galotni vai izskaņu, domu zīme nav jāņem vērā.
- I, II, III, VI, V, VI apzīmē 1. - 6. lietvārdu deklinācijas.
- I, II, III apzīmē 1. - 3. darbības vārdu konjugācijas.

1.1.4. Saistība ar citiem dokumentiem

Programmatūras prasību specifikācija ir kvalifikācijas darba „Latviešu valodas vārdu locītājs” sastāvdaļa.

Šis dokuments ir izstrādāts saskaņā ar standartu LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis”.

Šo specifikāciju ieteicams lietot kopā ar kādu latviešu valodas gramatikas rokasgrāmatu, kas sīkāk paskaidro gramatikas likumu nozīmi, izcelsmi un struktūru.

1.2. Vispārējais apraksts

1.2.1. Produkta perspektīva

Izstrādājamais produkts ir neatkarīga pirmkoda bibliotēka, kas tiks lietota kā citu programmatūras projektu sastāvdaļa un nodrošinās funkcionalitāti, kas saistīta ar vārdu locīšanu.

1.2.2. Produkta funkcijas

Programmatūras produktam jāspēj izlocīt trīs galveno vārdšķiru vārdi: lietvārdi, īpašības vārdi, darbības vārdi (un arī divdabji). Programmas produktam arī jāspēj darboties bez vārdnīcas, t.i., izlocīt jaunus un atvasinātus vārdus, kā arī jāatpazīst izņēmumi, kas nosakāmi pēc morfoloģijas likumiem un izņēmumi, kurus var pārskaitīt.

1.2.3. Lietotāja raksturiezīmes

Bibliotēka ir paredzēta tikai iekšējās programmatūras saskarnēs un tās lietotāji ir programmētāji, kuri izstrādā programmatūru, kas izmantos bibliotēkas funkcionalitāti. Lai izmantotu bibliotēkas funkcionalitāti, programmētājiem jābūt zināšanām par objektorientēto programmēšanu.

1.2.4. Pieņēmumi un atkarības

Tiek pieņemts, ka lietotā vide atbalsta un spēj attēlot pareizi latīņu alfabēta un latviešu burtus.

1.3. Funkcionālās prasības

Funkcionālās prasības atbilst latviešu valodas gramatikas likumiem. Šeit definētās prasības, izņēmumi un likumi ir ņemti no latviešu valodas gramatikas rokasgrāmatām un mācību literatūras. Lietvārdu un īpašības vārdu locīšanas tabulas ir ņemtas no rokasgrāmatas, savukārt darbības vārdu locīšanas tabulas ir veidotas pēc rokasgrāmatās atrodaļiem gramatikas likumiem, kas aprakstīti brīvā formā.

1.3.1. Lietvārdi

Lietvārdu locīšanai, zinot lietvārda formu vienskaitļa nominatīvā, dzimti un, vai tas ir sugas vārds vai personvārds, jāizloka vārds visos locījumos un skaitļos. Kopdzimtes lietvārdiem jānorāda tā dzimte, kurā lietvārds jāizloka.

1.3.1.1. Lokāmo lietvārdu locīšana

Ievads

Funkcija izloka deklinējamus I, II, III, IV, V un VI deklinācijas lietvārdus, ņemot vērā līdzskaņu miju un izņēmumus.

Ievade

Locīšanai nepieciešamā informācija

- Lietvārds vienskaitļa nominatīvā
- Deklinācija (I / II / III / IV / V / VI)
- Dzimte (vīriešu / sieviešu), ja deklinācija ir IV vai V
Ja dzimte nav norādīta, tiek pieņemts, ka tā ir sieviešu dzimte.
Ja ir vīriešu dzimte, loka pēc IV* vai V* deklinācijas parauga.
- Vārda iedalījums (sugas vārds / personvārds), ja deklinācija ir II.
Ja nav norādīts, tiek pieņemts, ka tas ir sugas vārds.

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)

Apstrāde

1. Vārda locīšana.
 - 1.1. Atmet locīšanas tabulā atrodamo galotni pēc norādītās deklinācijas vienskaitļa nominatīvā no lokāmā vārda beigām, iegūstot vārda celmu.
 - 1.2. Ja vārdam ir līdzskaņu mija, tad
 - 1.2.1. atmet mijas esošo daļu no vārda celma beigām,
 - 1.2.2. pievieno mijas jauno daļu vārda celma beigām.
 - 1.3. Vārda celma beigās pievieno locīšanas tabulā atrasto galotni pēc norādītās deklinācijas, locījuma un skaitļa.
2. Atgriež iegūto vārdu.

1.3.1.1.1. Locīšanas tabulas [1]

	Vienskaitlis			Daudzskaitlis
	I dekl.	II dekl.	III dekl.	I, II, III dekl.
Nominatīvs	-s, -š	-is / -s	-us	-i
Ģenitīvs	-a	-a / -s	-us	-u
Datīvs	-am	-im	-um	-iem
Akuzatīvs	-u	-i	-u	-us
Lokatīvs	-ā	-ī	-ū	-os

	IV dekl.		V dekl.		VI dekl.	
	vsk.	dsk.	vsk.	dsk.	vsk.	dsk.
Nominatīvs	-a	-as	-e	-es	-s	-is
Ģenitīvs	-as	-u	-es	-u	-s	-u
Datīvs	-ai -am*	-ām	-ei -em*	-ēm	-ij	-īm
Akuzatīvs	-u	-as	-i	-ēs	-i	-is
Lokatīvs	-ā	-ās	-ē	-es	-ī	-īs

1.3.1.1.2. Līdzskaņu mijas [1]

Locījumi	Mijas	Izņēmumi
II deklinācija		
1. Vienskaitļa ģenitīvs, ja vārds nav 1.3.1.4.1. sarakstā atrodamais izņēmums, 2. vienskaitļa ģenitīvs, ja vārds beidzas ar <i>suns</i> , 3. visi daudzskaitļa locījumi.	1. b → bj 2. m → mj 3. p → pj 4. v → vj 5. t → š 6. d → ž 7. c → č 8. dz → dž 9. s → š 10. z → ž 11. n → ņ 12. nn → ņņ 13. l → ļ 14. sn → šņ 15. zn → žņ 16. sl → šļ 17. zl → žļ 18. ln → ļņ	1. Vārdi, kas beidzas ar 1.1. <i>tētis</i> , 1.2. <i>viesis</i> . 2. Vārdi, kas beidzas ar 2.1. <i>-astis</i> , 2.2. <i>-jis</i> , 2.3. <i>-ķis</i> , 2.4. <i>-ģis</i> , 2.5. <i>-ris</i> , 2.6. <i>-skatis</i> . 3. Divzīlību personvārdi, kas beidzas ar 3.1. <i>-tis</i> , 3.2. <i>-dis</i> . 4. Uzvārdi, kas beidzas ar 4.1. <i>-skis</i> , 4.2. <i>-ckis</i> .

V deklinācija		
Daudzskaitļa ģenitīvs.	<ol style="list-style-type: none"> 1. b → bj 2. m → mj 3. p → pj 4. v → vj 5. c → č 6. t → š 7. d → ž 8. dz → dž 9. s → š 10. z → ž 11. n → ņ 12. l → ļ 13. sn → šņ 14. zn → žņ 15. kst → kš 	<ol style="list-style-type: none"> 1. Vārdi, kas beidzas ar <ol style="list-style-type: none"> 1.1. -aste, 1.2. -fe, 1.3. -ģe, 1.4. -ķe, 1.5. -mate, 1.6. -pēde. 2. Vārdi, kas beidzas ar <ol style="list-style-type: none"> 2.1. -ste, bet ne -kste. 3. Vārdi, kas beidzas ar <i>apaļmute</i>, <i>apšaude</i>, <i>balamute</i>, <i>balle</i>, <i>bāze</i>, <i>bise</i>, <i>bote</i>, <i>brīze</i>, <i>flote</i>, <i>fronte</i>, <i>gāze</i>, <i>gide</i>, <i>kase</i>, <i>kušete</i>, <i>mise</i>, <i>mute</i>, <i>pase</i>, <i>piešaude</i>, <i>planšete</i>, <i>rase</i>, <i>sarakste</i>, <i>šprote</i>, <i>takse</i>, <i>tirāde</i>.
VI deklinācija		
Daudzskaitļa ģenitīvs.	<ol style="list-style-type: none"> 1. v → vj 2. t → š 3. d → ž 4. s → š 5. z → ž 6. n → ņ 7. l → ļ 8. sn → šņ 9. st → š 	<ol style="list-style-type: none"> 1. Vārdi, kas beidzas ar <i>acs</i>, <i>ass</i>, <i>auss</i>, <i>balss</i>, <i>dakts</i>, <i>debess</i>, <i>dzelzs</i>, <i>kūts</i>, <i>maksts</i>, <i>pirts</i>, <i>šalts</i>, <i>takts</i>, <i>uts</i>, <i>uzacs</i>, <i>valsts</i>, <i>vēsts</i>, <i>zoss</i>, <i>žults</i>. 2. Vārdi, kas beidzas ar <i>brokastis</i>, <i>Cēsis</i>, <i>aktis</i>.

Izvade

Izlocīts vārds.

1.3.1.2. Nelokāmo lietvārdu locīšana

Ievads

Lai būtu saderība starp lokāmiem un nelokāmiem, funkcija „izloka” nelokāmos lietvārdus.

Ievade

Locīšanai nepieciešamā informācija

- Lietvārds

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)

Apstrāde

Atgriež doto vārdu.

Izvade

Dotais vārds.

1.3.1.3. Atgriezenisko lietvārdu locīšana

Ievads

Atgriezeniskos lietvārdus loka īpatnēji un tiem nav vairāku locījumu.

Ievade

Locīšanai nepieciešamā informācija

- Lietvārds vienskaitļa nominatīvā

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)

Apstrāde

1. Vārda locīšana.

1.1. Atmet no locīšanas tabulā atrodamajām galotnēm vienskaitļa nominatīvā atrasto galotni no lokāmā vārda beigām, iegūstot vārda celmu.

1.2. Ja locīšanas tabulā ir dota galotne pēc locījuma un skaitļa,

1.2.1. to pievieno vārda celma beigās,

1.2.2. atgriež iegūto vārdu.

1.3. Citādi atgriež tukšo vārdu.

1.3.1.3.1. Locīšanas tabula [1]

	-šanās		-umies		-tājies, -tājās, -ējies, -ējās, -ājies	
	vsk.	dsk.	vsk.	dsk.	vsk.	dsk.
Nominatīvs	-šanās	-šanās	-umies	-umies	-tājies, -tājās, ...	
Ģenitīvs	-šanās	-šanos				
Datīvs						
Akuzatīvs	-šanos	-šanās	-umos	-umos		
Lokatīvs						

Izvade

Izlocīts vārds vai tukšais vārds, ja vārdu nevarēja izlocīt.

1.3.1.4. Lietvārda locīšanas parauga noteikšana

Ievads

Ja ir zināms tikai fakts, ka vārds ir lietvārds, un iespējams arī tā dzimte, tiek mēģināts noteikt (bet bez garantijas), vai tas ir lokāms, nelokāms vai atgriezenisks lietvārds un attiecīgi tiek izlocīts. Var neobligāti norādīt dzimti, kas atvieglo noteikšanu, bet, ja tā nav norādīta, to cenšas uzminēt bez garantijas, ka tā būs uzminēta pareizi. Ir iespējams vārdam norādīt gramatiski nepareizu dzimti, tādā gadījumā vārds tik un tā tiek izlocīts, taču rezultāts var būt gramatiski nepareizs.

Ievade

Locīšanai nepieciešamā informācija

- Lietvārds vienskaitļa nominatīvā
- Dzimte (vīriešu / sieviešu) – nav obligāta
- Vārda iedalījums (sugas vārds / personvārds) – nav obligāts

Ja nav norādīts, tiek pieņemts, ka tas ir sugas vārds.

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)

Apstrāde

1. Nosaka deklināciju pēc deklinācijas noteikšanas tabulas.
 - 1.1. Ja ir noteikta deklinācija, doto vārdu izloka kā lokāmo vārdu ar 1.3.1.1. funkciju.
 - 1.2. Ja ir noteikts, ka vārds ir atgriezenisks lietvārds, to izloka kā atgriezenisko lietvārdu ar 1.3.1.3. funkciju.
 - 1.3. Citādi loka vārdu kā nelokāmu ar 1.3.1.2. funkciju.
2. Atgriež izlocīto vārdu.

1.3.1.4.1. Deklinācijas noteikšana

Vārda beigas	Vīr.dz.	Siev.dz.	Nav norādīta dzimte	Piezīmes
1. -s	I	VI	I	
2. -š	I		I	
3. -is	II	VI	II	
4. -us	III	VI	III	
5. -a	IV*	IV	IV	
6. -e	V*	V	V	
7. mēness, akmens, asmens, rudens, ūdens, zibens, suns, sāls	II	VI	II	II dekl. izņēmumi [1]
8. acs, asins, ass, azots, auss, avs, balss, birzs, blakts, cilts, dakts, debess, dūkst, dzelzs, govs, ilkss, izkapts, kārts, klēts, klints, krāsns, krūts, kūts, līksts, leceks, maksts, nāss, nakts, olekts, pāksts, palts, pils, pirts, sirds, smilts, šalts, takts, telts, uguns, uts, uzacs, valsts, vāts, vēsts, zivs, zoss, žults	I	VI	VI	B.Ceplītes un L.Ceplīša minētie VI deklinācijas lietvārdi [2]
9. brokastis, Cēss, durvis	I	VI	VI	daudzskaitlinieki pārvērsti vsk. [1]
10. acs, apakšgints, asins, ass, auss, akts, balss, bezvēsts, bikts, birzs, blakts, bārksts, cilts, dakts, debess, drāts, dzelzs, dzirksts, dūkst, govs, grunts, guns, ģikts, ģints, igvāts, ilkss, izkapts, klints, klēts, krants, krāsns, krūts, kvīts, kārts, kūts, laputs, leceks, maksts, nakts, nots, nāss, nīts, olekts, palts, pavēsts, pils, pirts, plīts, pulsts, pāksts, rūts, sirds, skrots, smilts, spīts, šalts, takts, telts, tāss, uguns, uts, uztakts, vakts, valsts, vants, versts, vāts, vēsts, zivs, zoss, žults	I	VI	VI	OpenOffice.org pareizrakstības pārbaudes vārdnīcā atrastie siev.dz. vārdi [3]
11. -šanās, -tājies, -tājās, -ējies, -ējās, -ājies, -umies	I	Atgr. lietv.	Atgr. lietv.	atgriezeniskie lietvārdi

Izvade

Izlocītais vārds.

1.3.2. Īpašības vārdi

Īpašības vārdu locīšanai, zinot īpašības formu pamata pakāpes vīriešu dzimtes vienskaitļa nominatīvā ar nenoteikto galotni, jāizloka vārds visos locījumos, skaitļos, dzimtēs un ar nenoteikto un noteikto galotni. No īpašības vārda jāizveido salīdzināmās pakāpes, kuras pēc tam var izlocīt kā īpašības vārdus.

1.3.2.1. Lokāmo īpašības vārdu locīšana

Ievads

Funkcija izloka lokāmos īpašības vārdus noteiktā locījumā, skaitlī, dzimtē un galotnē.

Ievade

Locīšanai nepieciešamā informācija

- Īpašības vārds pamata pakāpes vīriešu dzimtes vienskaitļa nominatīvā ar nenoteikto galotni

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)
- Dzimte (vīriešu / sieviešu)
- Galotne (noteiktā / nenoteiktā)

Apstrāde

1. Vārda locīšana.
 - 1.1. Atmet galotni -s vai -š no dotā vārda beigām, iegūstot vārda celmu.
 - 1.2. Pēc skaitļa / dzimtes / galotnes tabulā atrod galotni.
 - 1.3. Pievieno vārda celma beigās atrasto galotni.
2. Atgriež iegūto vārdu.

1.3.2.1.1. Locīšanas tabula [1]

	Vienskaitlis				Daudzskaitlis			
	Vīriešu dz.		Sieviešu dz.		Vīriešu dz.		Sieviešu dz.	
	Nenot.g.	Not.g.	Nenot.g.	Not.g.	Nenot.g.	Not.g.	Nenot.g.	Not.g.
Nominatīvs	-s, -š	-ais	-a	-ā	-i	-ie	-as	-ās
Ģenitīvs	-a	-ā	-as	-ās	-u	-o	-u	-o
Datīvs	-am	-ajam	-ai	-ajai	-iem	-ajiem	-ām	-ajām
Akuzatīvs	-u	-o	-u	-o	-us	-os	-as	-ās
Lokatīvs	-ā	-ajā	-ā	-ajā	-os	-ajos	-ās	-ajās

Izvade

Izlocītais vārds.

1.3.2.2. Nelokāmo īpašības vārdu locīšana

Ievads

Funkcija „izloka” nelokāmos īpašības vārdus noteiktā locījumā, skaitlī, dzimtē un galotnē.

Ievade

Locīšanai nepieciešamā informācija

- Īpašības vārds

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)
- Dzimte (vīriešu / sieviešu)
- Galotne (noteiktā / nenoteiktā)

Apstrāde

Atgriež doto īpašības vārdu.

Izvade

Izlocīts vārds, kas vienmēr sakrīt ar doto vārdu.

1.3.2.3. Pamata pakāpe

Ievads Izveido no īpašības vārda pamata pakāpi.

Ievade Īpašības vārds bez galotnes.

Apstrāde

1. Ja vārds sākas ar *vis*, tad to atmet.
2. Ja vārds beidzas ar *āk*, tad to atmet.
3. Atgriež atlikušo vārda daļu.

Izvade Īpašības vārds pamata pakāpē bez galotnes.

1.3.2.4. Pārākā pakāpe

Ievads Izveido no īpašības vārda pārāko pakāpi.

Ievade Īpašības vārds bez galotnes.

Apstrāde

1. Iegūst īpašības vārda pamata pakāpi ar funkciju 1.3.2.3.
2. Pievieno īpašības vārdam beigās *āk*.
3. Atgriež iegūto vārdu.

Izvade Īpašības vārds pārākajā pakāpē bez galotnes.

1.3.2.5. Vispārākā pakāpe

Ievads Izveido no īpašības vārda vispārāko pakāpi.

Ievade Īpašības vārds bez galotnes.

Apstrāde

1. Iegūst īpašības vārda pārāko pakāpi ar funkciju 1.3.2.4.
2. Pievieno īpašības vārdam sākumā *vis*.
3. Atgriež iegūto vārdu.

Izvade Īpašības vārds vispārākajā pakāpē bez galotnes.

1.3.3. Darbības vārdi

Darbības vārdu locīšanai, zinot nenoteiksmi, tagadnes celmu, pagātnes celmu, konjugāciju un grupu, jāizloka vārds visās izteiksmēs, vienkāršajos laikos, personās un skaitļos darāmajā kārtā.

Funkcijas pārskatāmības labad ir sadalītas pa izteiksmēm.

1.3.3.1. Īstenības izteiksmes

Darbības vārds īstenības izteiksmē izsaka darbību kā realitāti tagadnē, pagātnē vai nākotnē vai arī to noliedz. Šo attieksmi pret īstenību pauž darbības vārda laiku formas. Īpaša rādītāja īstenības izteiksmei nav. Īstenības izteiksmei ir visi seši laiki: vienkāršā tagadne, vienkāršā pagātne, vienkāršā nākotne, saliktā tagadne, saliktā pagātne un saliktā nākotne. [1]

Šeit aprakstīti ir tikai vienkāršie laiki, jo saliktie laiki tiek veidoti, kombinējot palīgdarbības vārdu un lokāmā darbības vārda divdabi.

1.3.3.1.1. Īstenības izteiksmes vienkāršā tagadne

Ievads Funkcija izloka darbības vārdu darāmās kārtas īstenības izteiksmes vienkāršajā tagadnē noteiktā personas formā un skaitlī.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē
- Darbības vārda tagadnes celms
- Darbības vārda konjugācija (I / II / III)
- Darbības vārda grupa (1-5)

Iegūstamās vārda formas aprakstošie parametri

- Personas forma (1. / 2. / 3.)
- Personas skaitlis (vienskaitlis / daudzskaitlis)

Apstrāde

1. Vārda locīšana.

1.1. Nosaka pieliekamās galotnes veidu:

1.1.1. Īsā galotne, ja darbības vārds nav III konjugācijas 1. grupas darbības vārds.

1.1.2. Garā galotne, ja darbības vārds ir III konjugācijas 1. grupas darbības vārds.

1.2. Ja nenoteiksme beidzas ar -t, vārds ir tiešs, citādi vārds ir atgriezenisks.

1.3. Pēc personas formas / skaitļa / vārda atgriezeniskuma / galotnes veida tabulā atrod galotni.

1.3.1. Tiešajiem darbības vārdiem vienskaitļa otrajā personā

1.3.1.1. Galotne -i ir

1.3.1.1.1. III konjugācijas darbības vārdiem,

1.3.1.1.2. I konjugācijas 5. grupas

1.3.1.1.3. I konjugācijas 3. grupas darbības vārdiem, kam tagadnes celms beidzas ar -d, -p vai -t.

1.3.1.2. Pārējiem darbības vārdiem nav galotnes

1.4. Tagadnes celmam veic līdzskaņu miju un tas būs vārda celms.

1.4.1. Ja darbības vārds ir II konjugācijas darbības vārds un persona ir 3. persona, tad no celma atmet -j.

1.5. Pievieno 1.4. solī iegūtajam vārda celmam 1.3. solī atrasto izskaņu.

2. Atgriež iegūto vārdu.

1.3.3.1.1.1. Locīšanas tabula

	Tiešs				Atgriezenisks			
	Īsā galotne		Garā galotne		Īsā galotne		Garā galotne	
	Vsk.	Dsk.	Vsk.	Dsk.	Vsk.	Dsk.	Vsk.	Dsk.
1. persona	-u	-am	-u	-ām	-os	-amies	-os	-āmies
2. persona	- (-i)	-at	- (-i)	-āt	-ies	-aties	-ies	-āties
3. persona	-		-a		-as		-ās	

1.3.3.1.1.2. Līdzskaņu mija [1]

1. 1. konjugācijas darbības vārdiem 2. personas vienskaitlī

1.1. bj → b

1.2. mj → m

1.3. pj → p

1.4. ļ → l

1.5. š → s

1.6. ž → z

1.7. k → c, ja nenoteiksme beidzas ar -kt vai -kties

1.8. g → dz, ja nenoteiksme beidzas ar -gt vai -gtiess

Izvade

Izlocītais vārds.

1.3.3.1.2. Īstenības izteiksmes vienkāršā pagātne

Ievads Funkcija izloka darbības vārdu darāmās kārtas īstenības izteiksmes vienkāršajā pagātnē noteiktā personas formā un skaitlī.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē
- Darbības vārda pagātnes celms.

Iegūstamās vārda formas aprakstošie parametri

- Personas forma (1. / 2. / 3.)
- Personas skaitlis (vienskaitlis / daudzskaitlis)

Apstrāde

1. Vārda locīšana.
 - 1.1. Ja nenoteiksme beidzas ar -t, vārds ir tiešs, citādi vārds ir atgriezenisks.
 - 1.2. Pēc personas formas / skaitļa / vārda atgriezeniskuma tabulā atrod galotni.
 - 1.3. Pievieno vārda pagātnes celmam 1.2. solī atrasto galotni.
2. Atgriež iegūto vārdu.

1.3.3.1.2.1. Locīšanas tabula

	Tiešs		Atgriezenisks	
	Vsk.	Dsk.	Vsk.	Dsk.
1. persona	-u	-ām	-os	-āmies
2. persona	-i	-āt	-ies	-āties
3. persona	-a		-ās	

Izvade

Izlocītais vārds.

1.3.3.1.3. Īstenības izteiksmes vienkāršā nākotne

Ievads Funkcija izloka darbības vārdu darāmās kārtas īstenības izteiksmes vienkāršajā nākotnē noteiktā personas formā un skaitlī.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē

Iegūstamās vārda formas aprakstošie parametri

- Personas forma (1. / 2. / 3.)
- Personas skaitlis (vienskaitlis / daudzskaitlis)

Apstrāde

1. Vārda locīšana.

- 1.1. Ja nenoteiksme beidzas ar -t, vārds ir tiešs, citādi vārds ir atgriezenisks.
- 1.2. Pēc personas formas / skaitļa / vārda atgriezeniskuma tabulā atrod izskaņu.
- 1.3. Nenoteiksmei no beigām atmet -t vai -ties, iegūstot nenoteiksmes celmu.
- 1.4. Nenoteiksmes celmam veic līdzskaņu miju un tas būs vārda celms.
- 1.5. Pievieno iegūtajam vārda celmam 1.2. solī atrasto izskaņu.

2. Atgriež iegūto vārdu.

1.3.3.1.3.1. Locīšanas tabula

	Tiešs		Atgriezenisks	
	Vsk.	Dsk.	Vsk.	Dsk.
1. persona	-šu	-sim	-šos	-simies
2. persona	-si	-siet	-sies	-sieties
3. persona	-s		-sies	

1.3.3.1.3.2. Līdzskaņu mija [1]

1. Ja ir I konjugācijas darbības vārds

1.1. Ja pagātnes celms beidzas ar -t

1.1.1. $s \rightarrow t$

1.2. Ja pagātnes celms beidzas ar -d

1.2.1. $s \rightarrow d$

1.3. Ja nenoteiksmes celms beidzas ar -s vai -z

1.3.1. $(nāk klāt) \rightarrow \bar{i}$

Izvade

Izlocītais vārds.

1.3.3.2. Atstāstījuma izteiksmes

Darbības vārds atstāstījuma izteiksmē izsaka darbību, par kuras realitāti runātājs neatbild. Atstāstījuma izteiksmi lieto, kad runātājs atstāsta no citiem dzirdēto.

1.3.3.2.1. Atstāstījuma izteiksmes vienkāršā tagadne

Ievads Funkcija izloka darbības vārdu darāmās kārtas atstāstījuma izteiksmes vienkāršajā tagadnē.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē
- Darbības vārda tagadnes celms

Apstrāde

1. Ja nenoteiksme beidzas ar -t, vārds ir tiešs, citādi vārds ir atgriezenisks.
2. Ja darbības vārds ir tiešs, atgriež tagadnes celmu + *-ot*
3. Ja darbības vārds ir atgriezenisks, atgriež tagadnes celmu + *-oties*.

Izvade

Izlocītais vārds.

1.3.3.2.2. Atstāstījuma izteiksmes vienkāršā nākotne

Ievads Funkcija izloka darbības vārdu darāmās kārtas atstāstījuma izteiksmes vienkāršajā nākotnē.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē

Apstrāde

1. No nenoteiksmes atmet -t vai -ties, iegūstot nenoteiksmes celmu.
2. Nenoteiksmes celmam pielieto īstenības izteiksmes vienkāršās nākotnes līdzskaņu miju no funkcijas 1.3.3.1.3, iegūstot nākotnes celmu.
3. Ja nenoteiksme beidzas ar -t, vārds ir tiešs, citādi vārds ir atgriezenisks.
4. Ja darbības vārds ir tiešs, atgriež nākotnes celmu + *-šot*
5. Ja darbības vārds ir atgriezenisks, atgriež nākotnes celmu + *-šoties*.

Izvade

Izlocītais vārds.

1.3.3.3. Vēlējuma izteiksmes

Darbības vārds vēlējuma izteiksmē izsaka vēlēšanos, lai darbība tiktu īstenota, vai arī norāda, ka darbība ir iespējama.

1.3.3.3.1. Vēlējuma izteiksmes vienkāršā tagadne

Ievads Funkcija izloka darbības vārdu darāmās kārtas atstāstījuma izteiksmes vienkāršajā nākotnē.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē.

Apstrāde

1. Ja nenoteiksme beidzas ar -t, vārds ir tiešs, citādi vārds ir atgriezenisks.
2. Atmet no nenoteiksmes -t vai -ties, iegūstot nenoteiksmes celmu.
3. Ja darbības vārds ir tiešs, atgriež nenoteiksmes celmu + *-tu*
4. Ja darbības vārds ir atgriezenisks, atgriež nenoteiksmes celmu + *-tos*.

Izvade

Izlocītais vārds.

1.3.3.4. Pavēles izteiksme

Ievads

Darbības vārds pavēles izteiksmē izsaka rosinājumu aicinājumu, lūgumu, aizliegumu, pavēli. Funkcija izloka darbības vārdu pavēles izteiksmē.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē.

Iegūstamās vārda formas aprakstošie parametri

- Personas forma (1. / 2. / 3.), kurā izlocīt vārdu.
- Personas skaitlis (vienskaitlis / daudzskaitlis), kurā izlocīt vārdu.

Apstrāde

1. Ja personas forma ir pirmā
 - 1.1. Atgriež īstenības izteiksmes vienkāršās nākotnes locījumu pirmās personas daudzskaitlī ar 1.3.3.1.3. funkciju.
2. Ja personas forma ir otrā
 - 2.1. Iegūst īstenības izteiksmes vienkāršās tagadnes otrās personas vienskaitļa locījumu ar funkciju 1.3.3.1.1.

- 2.2. Pielieto līdzskaņu miju
- 2.3. Ja personas skaitlis ir daudzskaitlis
 - 2.3.1. Ja darbības vārds ir tiešs, pievieno galotni -iet.
 - 2.3.2. Ja darbības vārds ir atgriezenisks, pievieno galotni -ieties.
- 2.4. Atgriež iegūto vārdu
- 3. Ja personas forma ir trešā
 - 3.1. Atgriež „lai „ + īstenības izteiksmes vienkāršās tagadnes vienskaitļa trešās personas locījumu ar funkciju 1.3.3.1.1.

1.3.3.4.1. Līdzskaņu mija [1]

- 1. Ja personas forma ir otrā un skaitlis ir daudzskaitlis
 - 1.1. Ja celms beidzas ar -i, to atmet.
 - 1.2. Ja celms beidzas ar -ies, to atmet.
- 2. Ja pagātnes celms beidzas ar -t
 - 2.1. $s \rightarrow t$
 - 2.2. $\check{z} \rightarrow t$
- 3. Ja pagātnes celms beidzas ar -d
 - 3.1. $s \rightarrow d$
 - 3.2. $z \rightarrow d$
- 4. Ja darbības vārds ir II konjugācijas darbības vārds un persona ir otrā un skaitlis ir daudzskaitlis
 - 4.1. $(n\grave{a}k\ kl\grave{a}t) \rightarrow j$

Izvade

Izlocītais vārds.

1.3.4. Divdabji

1.3.4.1. Darāmās kārtas tagadnes divdabīs

Ievads

Šo divdabi atvasina no darbības vārda tagadnes celma, pievienojot attiecīgo izskaņu. To loka kā īpašības vārdu.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārda tagadnes celms
- Darbības vārda pagātnes celms

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)
- Dzimte (vīriešu / sievietes)
- Galotne (noteiktā / nenoteiktā)

Apstrāde

1. Nosaka celmu

1.1. Pagātnes celms, ja darbības vārda tagadnes celms beidzas ar -š vai -ž.

1.2. Tagadnes celms, ja darbības vārda tagadnes celms nebeidzas ar -š vai -ž.

2. Pievieno izskaņu -ošs.

3. Loka kā īpašības vārdu ar funkciju 1.3.2.1.

Izvade

Izlocītais vārds.

1.3.4.2. Darāmās kārtas pagātnes divdabīs

Ievads

Šo divdabi atvasina no darbības vārda pagātnes celma. To daļēji loka kā īpašības vārdu.

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē
- Darbības vārda pagātnes celms

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)
- Dzimte (vīriešu / sievietes)
- Galotne (noteiktā / nenoteiktā)

Apstrāde

1. Ja nenoteiksme beidzas ar -t, tas ir tiešs darbības vārds un to loka pēc 1.3.4.2.1. tabulas.

2. Ja nenoteiksme beidzas ar -ties, tas ir atgriezenisks darbības vārds un to loka pēc 1.3.4.2.2. tabulas.

3. Pielieto līdzskaņu miju pagātnes celmam.

4. Locīšanās tabulā atrod galotni un to beigās pievieno pagātnes celmam.

4.1. Ja tabulā šūna ir tukša, tad locījuma nav un atgriež tukšu vārdu.

5. Atgriež vārdu.

1.3.4.2.1. Locīšanas tabula. * apzīmē attiecīgo tabulā 1.3.2.1.1. atrodamo galotni

	Vienskaitlis				Daudzskaitlis			
	Vīr.dz.		Siev.dz.		Vīr.dz.		Siev.dz.	
	Nenot.g.	Not.g.	Nenot.g.	Not.g.	Nenot.g.	Not.g.	Nenot.g.	Not.g.
Nominatīvs	-is	-ušais	-usi	-uši	-uš*			
Ģenitīvs	-uš*							
Datīvs								
Akuzatīvs								
Lokatīvs								

1.3.4.2.2. Locīšanas tabula. Tukša šūnā apzīmē, ka nav locījuma.

	Vienskaitlis		Daudzskaitlis	
	Vīr.dz.	Siev.dz.	Vīr.dz.	Siev.dz.
Nominatīvs	-ies	-usies	-ušies	-ušās
Ģenitīvs	-ušās		-ušos	
Datīvs				
Akuzatīvs	-ušos		-ušos	-ušās
Lokatīvs				

1.3.4.2.3. Līdzskaņu mija [1]

1. Nominatīvs, vienskaitlis

1.1. Vienskaitlis, vīriešu dzimte, nenoteiktā galotne

1.1.1. k → c

1.1.2. g → dz

1.2. Cits

1.2.1. c → k

1.2.2. dz → g

Izvade

Izlocītais vārds.

1.3.4.3. Ciešamās kārtas tagadnes divdabīs

Ievads

Šo divdabi atvasina no darbības vārda tagadnes celma, pievienojot attiecīgo izskaņu. To loka kā īpašības vārdu.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārda tagadnes celms
- Darbības vārda konjugācija (I / II / III)
- Darbības vārda grupa (1-5)

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)
- Dzimte (vīriešu / sievietes)
- Galotne (noteiktā / nenoteiktā)

Apstrāde

1. Tagadnes celmam pievieno izskaņu
 - 1.1. -ams, ja darbības vārds ir III konjugācijas 1. grupas darbības vārds.
 - 1.2. -āms, ja darbības vārds nav III konjugācijas 1. grupas darbības vārds.
2. Loka kā īpašības vārdu ar funkciju 1.3.2.1.

Izvade

Izlocītais vārds.

1.3.4.4. Ciešamās kārtas pagātnes divdabīs

Ievads

Šo divdabi atvasina no darbības vārda nenoteiksmes celma. To loka kā īpašības vārdu.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē

Iegūstamās vārda formas aprakstošie parametri

- Locījums (nominatīvs / ģenitīvs / datīvs / akuzatīvs / lokatīvs)
- Skaitlis (vienskaitlis / daudzskaitlis)
- Dzimte (vīriešu / sievietes)
- Galotne (noteiktā / nenoteiktā)

Apstrāde

1. Atmet no nenoteiksmes -t vai -ties.
2. Pievieno beigās izskaņu -ts.
3. Loka kā īpašības vārdu ar funkciju 1.3.2.1.

Izvade

Izlocītais vārds.

1.3.4.5. ***Dalēji lokāmais divdabis***

Ievads

Šo divdabi atvasina no darbības vārda nenoteiksmes celma.

Ievade

Locīšanai nepieciešamā informācija

- Darbības vārds nenoteiksmē

legūstamās vārda formas aprakstošie parametri

- Skaitlis (vienskaitlis / daudzskaitlis)
- Dzimte (vīriešu / sievietes)

Apstrāde

1. Atmet no nenoteiksmes -t vai -ties.
2. Ja darbības vārds nenoteiksmē beidzās ar -t, tas ir tiešs, citādi atgriezenisks.
3. Nosaka galotni pēc tabulas.
4. Nenoteiksmes celmam pievieno galotni.
5. Atgriež vārdu.

1.3.4.5.1. Locīšanas tabula

	Tiešs				Atgriezenisks	
	Vsk.		Dsk.			
	Vīr.dz.	Siev.dz.	Vīr.dz.	Siev.dz.	Vīr.dz.	Siev.dz.
Nomnatīvs	-dams	-dama	-dami	-damas	-damies	-damās

Izvade

Izlocītais vārds.

1.4. Nefunkcionālās prasības

1.4.1. Veiktspējas prasības

Tā kā paredzēts, ka bibliotēka bieži vien tiks izmantota, lai izlocītu vienu vārdu visos locījumos un iespējams lielu daudzumu vārdu secīgi, tad vārda locīšanai vienā locījumā nedrīkst patērēt vairāk kā 10 ms laika.

1.4.2. Ārējās saskarnes prasības

Bibliotēkas ārējai saskarnei ir jāapmierina šādas prasības:

- visa bibliotēkas funkcionalitāte (interfeiss), kas pieejama no ārienes, jāievieto atsevišķi nodalītā modulī, katras vārdšķiras (un „apakšvārdšķiras”) funkcionalitātei jābūt atsevišķi nodalītai tā, lai nebūtu nosaukumu konfliktu starp vārdšķirām,
- bibliotēka nedrīkst izmantot citas bibliotēkas un nedrīkst būt atkarīga no darbojošās vides, drīkst izmantot tikai programmēšanas valodas un standartbibliotēkas funkcionalitāti (interfeisu).

2. Programmatūras projektējuma apraksts

2.1. Ievads

2.1.1. Dokumenta nolūks

Programmatūras projektējuma apraksts ir paredzēts, lai aprakstītu kā izstrādājamās latviešu valodas vārdu locīšanas bibliotēkas programmatūras prasību specifikācijā minētās prasības tiks realizētas projektā. Dokuments radīts, lai pārveidotu prasības izstrādei parocīgākā formā, kas nodrošinātu bibliotēkas pirmkoda pārskatāmību un lasāmību, paplašinātību, kā arī testējamību.

2.1.2. Darbības sfēra

Izstrādājamais projekts ir neatkarīga bibliotēka, kas nodrošina latviešu valodas vārdu locīšanu. Programmatūras produkta uzdevums ir veikt visas nepieciešamās darbības, lai izveidotu nepieciešamo vārda formu.

2.1.3. Definīcijas, saīsinājumi un apzīmējumi

Šajā dokumentā lietotie projektam specifiskie jēdzieni, saīsinājumi un apzīmējumi ir definēti darba 1. nodaļas 1.1.3. sadaļā.

2.2. Saistība ar citiem dokumentiem

Programmatūras projektējuma apraksts ir kvalifikācijas darba „Latviešu valodas vārdu locītājs” sastāvdaļa.

Šis dokuments ir lietojams kopā ar latviešu valodas vārdu locītāja bibliotēkas programmatūras prasību specifikāciju.

Dokuments ir izstrādāts saskaņā ar standartu LVS 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”.

2.3. Dekompozīcijas apraksts

2.3.1. Ievads

Objektorientētās programmēšanas principi ir ļoti ērti attiecināmi un latviešu valodas gramatiku.

Vārds ir attēlots kā klase. Vārdam ir vairāki raksturojoši elementi (šajā projektā interesējošās lietas ir celms un galotne) un tie ir attēloti kā objekta īpašības (properties). Šajā bibliotēkā atsevišķam vārdam nav pielietojuma, tāpēc klase ir abstrakta.

Lietvārdi, īpašības vārdi un darbības vārdi arī ir vārdi ar vairākām kopīgām un vairākām atšķirīgām īpašībām. Tāpēc katras vārdšķiras vārdiem tiek veidota atsevišķa klase, kura manto klases Vārds īpašības.

Lietvārdus var locīt pēc deklinācijas, locījuma un skaitļa. Tā kā locīšana ir darbība, tad Lietvārds klasei ir metode Locīt, kas pēc parametriem locījums un skaitlis darina vārda formu šajā locījumā un skaitlī. Locījumu ir noteikts skaits un katram locījumam ir nosaukums, tāpēc tie ir attēloti kā vērtību saraksts (enum). Katras deklinācijas lietvārdi ir lokāmi citādāk, tāpēc katram locīšanas paraugam ir sava klase, kas manto īpašības no Lietvārds klases (polimorfisms). Katra locīšanas parauga klase pārraksta (override) Lietvārds klases abstraktās galotņu metodes un var mainīt lietvārdiem raksturojošo līdzskaņu miju, kas attēlota kā virtuāla LīdzskaņuMija metode. Sīkāk par to aprakstīts 2.5.2. Ja kādreiz būs nepieciešams pamainīt kādu locīšanas paraugu, neietekmējot esošo risinājumu, bet izveidojot jaunu, tad to varēs darīt pārrakstot nepieciešamās metodes vai arī izmantojot Decorator pattern [8]. Lietvārdu locīšanas parauga noteikšanai tiek izmantots Factory pattern [9].

Līdzīgi ir projektēti īpašības vārdi un darbības vārdi.

Divdabji ir vārdi, kas darināti no darbības vārdiem, bet lokāmi kā īpašības vārdi. Tāpēc divdabji manto īpašības vārda īpašības un lokāmais īpašības vārds konstruktorā tiek darināts no darbības vārda, izmantojot tā īpašības.

2.3.2. Entīju dekompozīcija

2.3.2.1. Vārdi

Nosaukums	Tips	Apraksts	Prasības
Vārds	Abstrakta klase	Glabā vārda celmu un galotni.	visas
BeidzasAr	Funkcija	Pārbauda, vai vārds beidzas ar kādu no pārbaudāmajām galotnēm.	visas
AtmestGalotni	Funkcija	Ja vārds beidzas ar galotni, to atmet, citādi neko nemaina.	visas
Apcirpt	Funkcija	Ja vārds sākas vai beidzas ar kādu tekstu, to noņem, citādi neko nemaina.	1.3.2.3., 1.3.2.4., 1.3.2.5.
LīdzskaņuMija	Funkcija	Ja vārds beidzas ar nomaināmo līdzskani, to aizvieto ar nomainīto līdzskani.	1.3.1.1.
Skaitlis	Vērtības	Uzskaita gramatiskos skaitļus.	1.3.1.*
Dzimte	Vērtības	Uzskaita gramatiskās dzimtes.	1.3.1.4.

2.3.2.2. Lietvārdi

Nosaukums	Tips	Apraksts	Prasības
Lietvārds	Abstrakta klase	Apraksta lietvārda īpašības, kā arī izveido jaunu lietvārdu, nosakot, pēc kāda parauga locīt.	1.3.1.1., 1.3.1.2., 1.3.1.3., 1.3.1.4.
Pirmās Deklinācijas Lietvārds	Klase	Izloka lietvārdu pēc I dekl. parauga.	1.3.1.1.
Otrās Deklinācijas Lietvārds	Klase	Izloka lietvārdu pēc II dekl. parauga.	1.3.1.1.
Otrās Deklinācijas Personvārda Lietvārds	Klase	Izloka lietvārdu pēc II dekl. parauga, ņemot vērā personvārdu izņēmumus.	1.3.1.1.
Trešās Deklinācijas Lietvārds	Klase	Izloka lietvārdu pēc III dekl. parauga.	1.3.1.1.
Ceturtās Deklinācijas Lietvārds	Klase	Izloka lietvārdu pēc IV dekl. parauga.	1.3.1.1.
Piektās Deklinācijas Lietvārds	Klase	Izloka lietvārdu pēc V dekl. parauga.	1.3.1.1.
Sestās Deklinācijas Lietvārds	Klase	Izloka lietvārdu pēc VI dekl. parauga.	1.3.1.1.
Nelokāms Lietvārds	Klase	Izloka nelokāmu lietvārdu.	1.3.1.2.

Atgriezenisks Lietvārds	Klase	Izloka atgriezenisku lietvārdu.	1.3.1.3.
Locījums	Vērtības	Uzskaita lietvārda locījumus.	1.3.1.1., 1.3.1.2., 1.3.1.3., 1.3.1.4.

2.3.2.3. Īpašības vārdi

Nosaukums	Tips	Apraksts	Prasības
Īpašības Vārds	Klase	Izloka īpašības vārdu un veido salīdzināmās pakāpes.	1.3.2.1., 1.3.2.3., 1.3.2.4., 1.3.2.5.
Nelokāms Īpašības Vārds	Klase	Izloka nelokāmu īpašības vārdu.	1.3.2.2.
Locījums	Vērtības	Uzskaita īpašības vārda locījumus.	1.3.2.1., 1.3.2.2.
Galotne	Vērtības	Uzskaita īpašības vārda galotnes.	1.3.2.1., 1.3.2.2.

2.3.2.3. Darbības vārdi

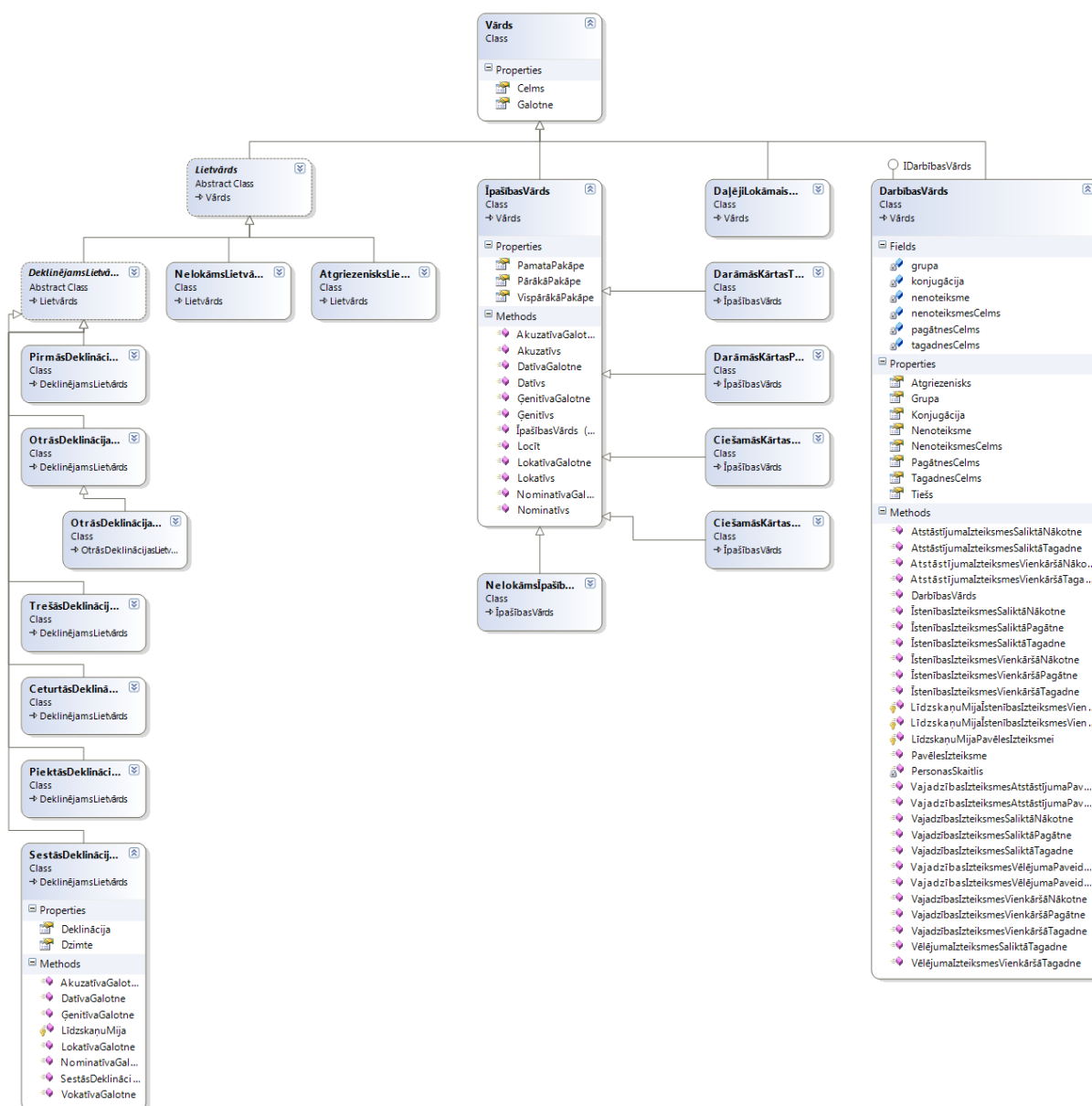
Nosaukums	Tips	Apraksts	Prasības
Darbības Vārds	Klase	Izloka īpašības vārdu un veido salīdzināmās pakāpes.	1.3.3.*
Konjugācija	Vērtības	Uzskaita konjugācijas.	1.3.3.1.1.
Persona	Vērtības	Uzskaita darbības vārda personas.	1.3.3.1.*,

2.3.2.4. Darbības vārdi / divdabji

Nosaukums	Tips	Apraksts	Prasības
Darāmās Kārtas Tagadnes Divdabis	Klase	Izloka darāmās kārtas tagadnes divdabi kā īpašības vārdu.	1.3.4.1.
Darāmās Kārtas Pagātnes Divdabis	Klase	Izloka darāmās kārtas pagātnes divdabi kā īpatnēju īpašības vārdu.	1.3.4.2.
Ciešāmās Kārtas Tagadnes Divdabis	Klase	Izloka ciešāmās kārtas tagadnes divdabi kā īpašības vārdu.	1.3.4.3.
Ciešāmās Kārtas Pagātnes Divdabis	Klase	Izloka ciešāmās kārtas pagātnes divdabi kā īpašības vārdu.	1.3.4.4.
Daļēji Lokāmais Divdabis	Klase	Izloka daļēji lokāmo divdabi vienā locījumā.	

2.4. Atkarību projektējums

Zemāk dotajā entītiju atkarību projektējumā no entītijas a uz entītiju b ir šķautne, ja a manto īpašības (un tātad ir atkarīga) no b.



2.5. Detalizētais projektējums

2.5.1. Vārdi

2.5.1.1. Vārds

Vārds ir abstrakta klase, kuru manto visas pārējās vārdšķiru klases. Vārds glabā vārda celmu un galotni. Šī klase tika ieviesta, jo vārda celms un galotne jāatceras gandrīz visiem locīšanas paraugiem

Nākotnē varētu paplašināt šīs klases funkcionalitāti ar, piemēram, zibļu skaitīšanu, automātisku vārda dalīšanu vārdu daļās, serializēšanu u.tml. funkcionalitāti, kas būtu uzreiz automātiski pieejama visām vārdšķirām.

Šī klase nebūt nav nepieciešama. Alternatīva ir šādas klases nelietošana, un vārda celms un galotne tiek noteikti pie katras locīšanas vai arī tiek glabāti katrā paraugā atsevišķi.

2.5.1.2. BeidzasAr

Iebūvētās teksta apstrādes funkcijas satur funkciju, kas pārbauda, vai teksta virkne beidzas ar kādu citu virkni. Lielākajā daļā locīšanas nācās pārbaudīt, vai vārds beidzas ar kādu no vairākām iespējamām galotnēm, tāpēc tika ieviesta palīgfunckcija.

Ievade

- Teksta virkne ar vārdu
- Teksta virknes ar galotnēm

Apstrāde

Atgriež patiess, ja vārds beidzas ar kādu no galotnēm.

Izvade

Patiess/nepatiesss.

2.5.1.3. AtmestGalotni

Iebūvētajām teksta apstrādes funkcijām katrreiz nācās padot galotnes garumu, lai to varētu noņemt no vārda beigām. Lai uzlabotu koda lasāmību, tika ieviesta palīgfunckcija, kurai jāpadod tikai vārds un galotne.

Ievade

- Teksta virkne ar vārdu
- Teksta virkne ar galotni

Apstrāde

Ja vārds beidzas ar galotni, to atmet un atgriež vārdu bez galotnes.

Izvade

Vārds ar noņemtu galotni.

2.5.1.4. Apcirpt

Iebūvētajām teksta funkcijām, lai noņemtu liekos simbolus no teksta virknes sākuma vai beigām, nācās padot sarakstu ar simboliem, bet ne teksta virknēm. Tāpēc nācās ieviest palīgfunckciju.

Ievade

- Teksta virkne ar vārdu
- Teksta virkne lieko tekstu
- Vai noņemt no sākuma (pēc noklusējuma nē)
- Vai noņemt no beigām (pēc noklusējuma nē)

Apstrāde

1. Ja vārds sākas ar lieko tekstu un jānoņem no sākuma, to noņem.
2. Ja vārds beidzas ar lieko tekstu un jānoņem no beigām, to noņem.
3. Atgriež atlikušo vārdu.

Izvade

Vārds ar noņemtu lieko tekstu.

2.5.1.5. LīdzskaņuMija

Ja vārds beidzas ar līdzskaņu mijas esošo daļu, to atmet un aizvieto ar jaunu daļu, citādi atgriež nemainītu vārdu.

Ievade

- Teksta virkne ar vārdu
- Teksta virkne ar esošo līdzskaņu mijas daļu
- Teksta virkne ar jauno līdzskaņu mijas daļu

Apstrāde

1. Ja vārds beidzas ar esošo daļu,
 - 1.1. to noņem no vārda beigām,
 - 1.2. pievieno vārda beigās jauno daļu.
2. Atgriež vārdu.

Izvade

Vārds ar līdzskaņu miju vai nemainīts vārds, ja mijas nebija.

2.5.1.6. Skaitlis

Lai varētu locīšanas funkcijai padot tikai derīgu skaitli un būtu pārskatāmi visi iespējamie skaitļi, kā arī būtu iespējams izlocīt vārdu visos skaitļos, izejot cauri skaitļu sarakstam, ievieš skaitļa vērtību sarakstu:

- Vienskaitlis = 1 (galvenais)
- Daudzskaitlis = 2

2.5.1.7. Dzimte

Lai varētu locīšanas funkcijai padot tikai derīgu dzimti un būtu pārskatāmas visas iespējamās dzimtes, kā arī būtu iespējams izlocīt vārdu visās dzimtēs, izejot cauri dzimtu sarakstam, ievieš dzimtu vērtību sarakstu:

- Vīriešu dzimte = 1 (galvenais)
- Sieviešu dzimte = 2

2.5.2. Lietvārdi

Lai atvieglotu testējamību, koda pārskatāmību, uzturēšanu un ievērotu objektorientētās programmēšanas principus, lietvārdu locīšana ir iedalīta vairākās klasēs, kur katra klase atbild par vienas noteiktas lietas veikšanu.

Katra lietvārdu klase ir izloka lietvārdu pēc citādāka parauga. Katra klase ir atbildīga par galotnes noteikšanu locījumam un skaitlim, kā arī līdzskaņu mijas veikšanu vārda celmam vajadzīgajos locījumos.

Alternatīvs risinājums būtu izveidot vienu klasi (vai funkciju) visiem lietvārdiem, taču tādā gadījumā klase saturētu ļoti daudz loģikas un būtu grūtāk pārvaldāms. Esošajā risinājumā, lai pievienotu jaunu locīšanas paraugu, atliek izveidot jaunu klasi, taču vienas klases risinājumā paraugs būtu jāpievieno šai klasei, padarot to lielāku un iespējams nepārskatāmāku.

2.5.2.1. Lietvārds

Lietvārds ir abstrakta klase, kas tikai definē locīšanas metodes, kuras pārraksta mantotās klases.

Klasei ir publiskas abstraktas galotņu metodes katram locījumam, kas pieņem vienu parametru – skaitli. Mantotās klases pārraksta šīs metodes un pēc locīšanas tabulas nosaka galotni locījumam un skaitlim un to atgriež.

Klasei ir aizsargāta virtuāla līdzskaņu mijas metode, kas pieņem divus parametrus – locījumu un skaitli. Ja mantotajai klasei šajā locījumā un skaitlī ir līdzskaņu mija, tā pārraksta šo metodi un veic saglabātajam vārda celmam līdzskaņu miju un to atgriež. Citādi metode atgriež nemainītu vārda celmu.

Klasei ir publiskas metodes katram locījumam, kuras nav paredzēts (bet nav aizliegts) pārrakstīt. Klase veic vārda locīšanu tā: izsauc līdzskaņu mijas metodi (kas atgrieztis nemainītu celmu, ja līdzskaņu mijas nav) un iegūst vārda celmu, izsauc locījuma galotņu metodi, kas atgrieztis galotni vajadzīgajam locījumam, un to pievieno vārda celma beigās. Iegūto vārdu atgriež.

Klasei ir papildus locīšanas funkcija, lai apmierinātu prasības, kas pieņem divus parametrus – locījumu un skaitli. Funkcija izsauc attiecīgo locījuma metodi ar skaitli kā parametru.

Klases loceklis	Nosakums	Ievade	Izvade
Aizsargāts konstruktors	Lietvārds	Celms	
Publiska, abstrakta metode	Nominatīva Galotne	Skaitlis	Galotne skaitļa nominatīvā
Publiska, abstrakta metode	Ģenitīva Galotne	Skaitlis	Galotne skaitļa ģenitīvā
Publiska, abstrakta metode	Datīva Galotne	Skaitlis	Galotne skaitļa datīvā
Publiska, abstrakta metode	Akuzatīva Galotne	Skaitlis	Galotne skaitļa akuzatīvā
Publiska, abstrakta metode	Lokatīva Galotne	Skaitlis	Galotne skaitļa lokatīvā
Aizsargāta, virtuāla metode	Līdzskaņu Mija	Locījums, skaitlis	Nemainīts celms vai celms ar veiktu līdzskaņu miju
Publiska metode	Locīt	Locījums, Skaitlis	Vārds skaitļa locījumā
Publiska metode	Nominatīvs; Ģenitīvs; ...	Skaitlis	Vārds skaitļa locījumā

2.5.2.2. Deklinējamie lietvārdi

Attiecas uz: Pirmās Deklinācijas Lietvārds, Otrās Deklinācijas Lietvārds, Otrās Deklinācijas Personvārda Lietvārds, Trešās Deklinācijas Lietvārds, Ceturtās Deklinācijas Lietvārds, Piektās Deklinācijas Lietvārds, Sestās Deklinācijas Lietvārds.

Klases manto īpašības no abstraktās Lietvārds klases.

Klases konstruktorā tiek pārbaudīts, vai vārds beidzas ar attiecīgās deklinācijas vienskaitļa nominatīva galotni vai galotnēm. Ja ir iespējamās vairākas galotnes, galotni saglabā, jo tā būs nepieciešama, lai atgrieztu vienskaitļa nominatīva galotni. Tiek saglabāta arī atlikušā vārda daļa kā celms.

Klases pārraksta iemantotās galotņu metodes un atgriež galotni no locīšanas tabulas pēc deklinācijas, locījuma un skaitļa.

Klases pārraksta iemantoto līdzskaņu mijas metodi, ja kādā no locījumiem vai skaitļiem ir līdzskaņu mija.

2.5.2.3. *NelokāmsLietvārds*

Klase manto īpašības no abstraktās Lietvārds klases. Klase konstruktorā pieņem vārdu un to saglabā kā celmu. Klase galotņu locījuma metodēs atgriež tukšu galotni.

2.5.2.4. *AtgriezenisksLietvārds*

Klase manto īpašības no abstraktās Lietvārds klases. Klase pārraksta galotņu metodes. Ja nav kāda locījuma, pārraksta līdzskaņu mijas metodi un tur atgriež tukšu vārdu.

2.5.2.5. *Locījums*

Lai varētu locīšanas funkcijai padot tikai derīgu locījumu un būtu pārskatāmi visi iespējamie locījumi, kā arī būtu iespējams izlocīt vārdu visos locījumos, izejot cauri locījumu sarakstam, ievieš locījumu vērtību sarakstu:

- Nominatīvs = 1 (galvenais)
- Ģenitīvs = 2
- Datīvs = 3
- Akuzatīvs = 4
- Lokatīvs = 5

2.5.3. Īpašības vārdi

2.5.3.1. *ĪpašībasVārds*

Īpašības vārda klase konstruktorā pieņem ievadvārdu, atmet galotni un saglabā vārda celmu un atmesto galotni. Locīšanas metode saglabātajam vārda celmam pievieno prasītā locījuma galotni. Pamata pakāpe tiek veidota no saglabātā celma un pārējās pakāpes tiek veidotas, balstotise uz pamata pakāpi.

2.5.3.1.1. *ĪpašībasVārds konstruktors*

Nosaukums ĪpašībasVārds.ĪpašībasVārds

Tips Konstruktors

Ievade Teksta virkne ar vārdu

Apstrāde

1. Ja vārds nebeidzas ar -s vai -š, tad paziņo par nederīgu ievadi.
2. Atmet no vārda -s vai -š.
3. Saglabā atlikušo vārdu un atmesto galotni.

2.5.3.1.2. ĪpašībasVārds.locīšana

Nosakums ĪpašībasVārds.Locīt

Tips Metode

Ievade Locījums, dzimte, skaitlis, galotne

Apstrāde

1. Pēc ievades locīšanas tabulā atrod galotni.
2. Pievieno galotni saglabātajam vārdam un to atgriež.

Izvade

Teksta virkne ar vārda formu.

2.5.3.1.3. ĪpašībasVārds.PamataPakāpe

Nosakums ĪpašībasVārds.PamataPakāpe

Tips Īpašība

Apstrāde

1. No saglabātā vārda sākuma noņem „vis”, ja tāds ir.
2. Pēc tam noņem no beigām „āk”, ja tāds ir.
3. Atgriež iegūto vārdu.

Izvade

Teksta virkne ar vārda formu.

2.5.3.1.4. ĪpašībasVārds.PārākāPakāpe

Nosakums ĪpašībasVārds.PārākāPakāpe

Tips Īpašība

Apstrāde

1. ĪpašībasVārds.PamataPakāpe rezultātam pievieno beigās „āk”.
2. Atgriež iegūto vārdu.

Izvade

Teksta virkne ar vārda formu.

2.5.3.1.5. ĪpašībasVārds.VispārākāPakāpe

Nosakums ĪpašībasVārds.VispārākāPakāpe

Tips Īpašība

Apstrāde

1. ĪpašībasVārds.PārākāPakāpe rezultātam pievieno sākumā „vis”.
2. Atgriež iegūto vārdu.

Izvade

Teksta virkne ar vārda formu.

2.5.3.2. *NelokāmsĪpašībasVārds*

2.5.3.1.1. *NelokāmsĪpašībasVārds konstruktors*

Nosaukums NelokāmsĪpašībasVārds. NelokāmsĪpašībasVārds

Tips Konstruktors

Ievade Teksta virkne ar vārdu

Apstrāde

Saglabā ievades vārdu.

2.5.3.1.2. *NelokāmsĪpašībasVārds locīšana*

Nosaukums NelokāmsĪpašībasVārds. Locīt

Tips Metode

Ievade Locījums, dzimte, skaitlis, galotne

Apstrāde

Atgriež saglabāto vārdu.

Izvade

Teksta virkne ar vārda formu.

2.5.3.3. *Locījums*

Lai varētu locīšanas funkcijai padot tikai derīgu locījumu un būtu pārskatāmi visi iespējamie locījumi, kā arī būtu iespējams izlocīt vārdu visos locījumos, izejot cauri locījumu sarakstam, ievieš locījumu vērtību sarakstu:

- Nominatīvs = 1 (galvenais)
- Ģenitīvs = 2
- Datīvs = 3
- Akuzatīvs = 4
- Lokatīvs = 5

2.5.3.4. *Galotne*

Lai varētu locīšanas funkcijai padot tikai derīgu galotnes veidu un būtu pārskatāmi visi iespējamie galotņu veidi, kā arī būtu iespējams izlocīt vārdu ar visām galotnēm, izejot cauri galotņu sarakstam, ievieš galotņu vērtību sarakstu:

- NenoteiktāGalotne = 1 (galvenais), NoteiktāGalotne = 2

2.5.4. Darbības vārdi

Darbības vārdu apzīmē viena klase DarbībasVārds. Klasē katrai izteiksmei un laikam tika piešķirta atsevišķa metode, jo katras metodes apstrādes lielums bija pietiekoši liels. Klasei ir arī īpašības ar nemainīgajām aprakstošajām formām, kuras izmanto locīšanā izteiksmēs.

Lai gan būtu bijis iespējams darbības vārda klasi iedalīt smalkāk TiešaisDarbībasVārds un AtgriezeniskaisDarbībasVārds un tas palīdzētu izvairīties no dažiem sazarojumiem, tas radītu daudz dublikācijas, tāpēc tika nolemts izmantot tikai vienu klasi un atgriezeniskumu saglabāt kā īpašību.

Klases konstruktors pieņem visus locīšanai nepieciešamos parametrus: nenoteiksmi, tagadnes celmu, pagātnes celmu, konjugāciju un grupu. Visi šie parametri tiek padarīti kā klases īpašības.

Katrai izteiksmei un laikam ir sava metode, kurā pēc personas formas un skaitļa (ja tas nepieciešams), izvēlas vajadzīgo celmu, galotni un tos kopā atgriež.

Saliktajos laikos, kur nepieciešams palīgdarbības vārds, tiek izveidots jauns darbības vārds un divdabis, tie attiecīgi izlocīti un atgriezti kopā, atdalīti ar atstarpī.

Līdzskaņu mijas ir atdalītas atsevišķās metodēs, taču netiek pielietotas katrā izteiksmē un laikā, kā tas ir lietvārdos.

Nekārtnie darbības vārdi vienkārši manto īpašības no DarbībasVārds un pārraksta atšķirīgās izteiksmes un laikus.

Klases loceklis	Nosaukums	Ievade	Izvade
Īpašība	Nenoteiksme	Nenoteiksme	Nenoteiksme
Īpašība	Nenoteiksmes Celms	Nenoteiksme	Nenoteiksmes celms
Īpašība	Tagadnes Celms	Celms vai tagadnes forma vsk. 1. personā	Tagadnes celms
Īpašība	Pagātnes Celms	Celms vai pagātnes forma vsk. 1. personā	Pagātnes celms
Īpašība	Tiešs	Nenoteiksme	Vai darbības vārds ir tiešs
Īpašība	Atgriezenisks	Nenoteiksme	Vai darbības vārds ir atgriezenisks

Publiska metode	Īstenības Izteiksmes Vienkāršā Tagadne	Persona, skaitlis	Izlocīts vārds
Publiska metode	Īstenības Izteiksmes Vienkāršā Pagātne	Persona, skaitlis	Izlocīts vārds
Publiska metode	Īstenības Izteiksmes Vienkāršā Nākotne	Persona, skaitlis	Izlocīts vārds
Publiska metode	Atstāstījuma Izteiksmes Vienkāršā Tagadne		Izlocīts vārds
Publiska metode	Atstāstījuma Izteiksmes Vienkāršā Nākotne		Izlocīts vārds
Publiska metode	Vēlējuma Izteiksmes Vienkāršā Tagadne		Izlocīts vārds
Publiska metode	Pavēles Izteiksme	Persona, Skaitlis	Izlocīts vārds bez izsaukuma zīmes

2.5.4.1. Konjugācija

Lai varētu locīšanas funkcijai padot tikai derīgu konjugāciju un būtu pārskatāmas visas iespējamās konjugācijas, ievieš konjugāciju vērtību sarakstu:

- I = 1
- II = 2
- III = 3

2.5.4.2. Persona

Lai varētu locīšanas funkcijai padot tikai derīgu personu un būtu pārskatāmas visas iespējamās locījumas, kā arī būtu iespējams izlocīt vārdu visās personās, izejot cauri personu sarakstam, ievieš personu vērtību sarakstu:

- Pirmā = 1 (*vai arī EsMēs*)
- Otrā = 2 (*vai arī TuJūs*)
- Trešā = 3 (*vai arī ViņiViņas*)

2.5.5. Divdabji

Līdzīgi kā pārējām vārdšķirām, katram divdabim ir sava klase. Tie divdabji, kas tiek locīti kā īpašības vārdi, manto īpašības no īpašības vārda un celmu izveido konstruktorā. Daļēji lokāmais divdabis papildus pārraksta dažas galotņu metodes.

3. Testēšanas dokumentācija

Lai pārbaudītu locītāja pareizību, tika izveidoti vienībtesti, kur katrs tests sastāv no pareizi izlocīta vārda. Testi tika veidoti, katru vārdu pašrocīgi izlokot visos locījumos.

Novecojušiem vārdiem locījumi tika pārbaudīti ar vārdnīcu un gramatikas rokasgrāmatu.

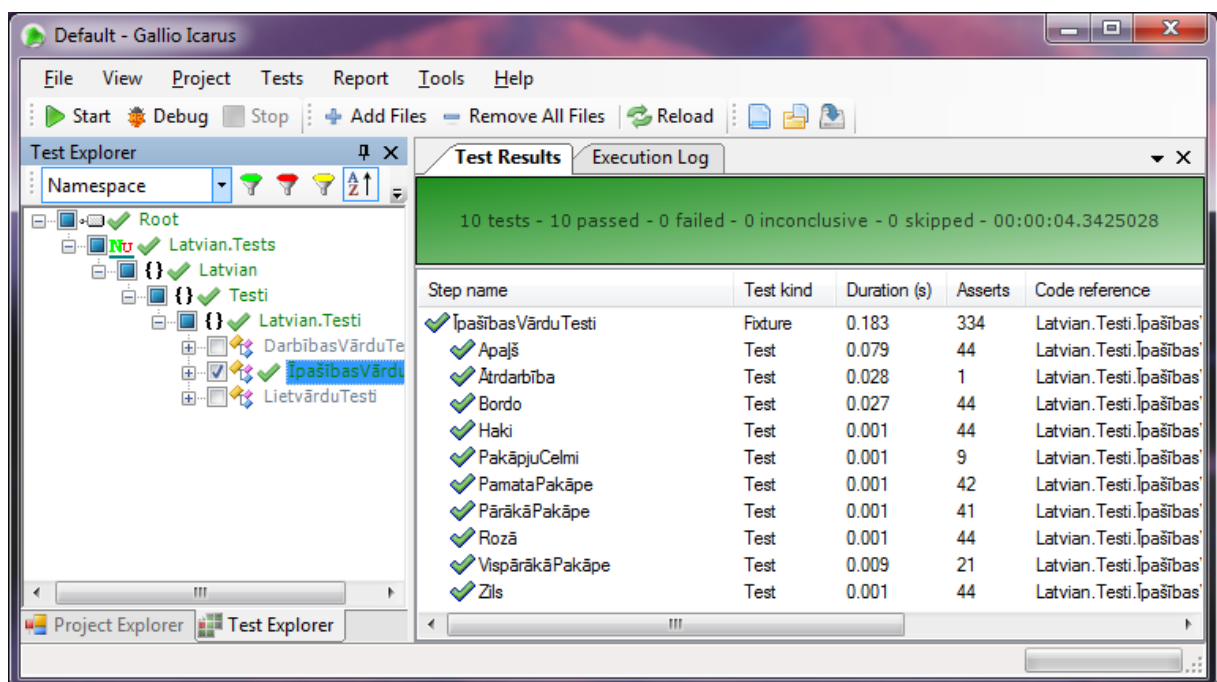
Testi regulārajiem vārdiem tika veidoti tā, lai pārliecinātos, ka rezultāti ir atbilstoši locīšanas tabulām. Pārējie testi domāti, lai pārbaudītu, vai locītājs spēj atpazīt un pareizi izlocīt izņēmumus. Tā kā regulāri lokāmo vārdu ir visvairāk un izņēmumi ir salīdzinoši maz, tad vēl neatpazītie izņēmumi būtiski nepadara bibliotēku par nelietojamu.

Vienībtesti tika veidoti ar NUnit [10] ievāru un testēti ar Gallio [11], kam ir ērtāks pārvaldīšanas un atskaišu rīks.

3.1. Īpašības vārdu testi

Īpašības vārdiem izņēmumu nav, tāpēc tika veidoti tikai divi piemēri īpašības vārdiem ar galotni -s, galotni -š un arī trīs piemēri nelokāmajiem īpašības vārdiem. Papildus tika testēta pakāpju veidošana un veikspējas prasības.

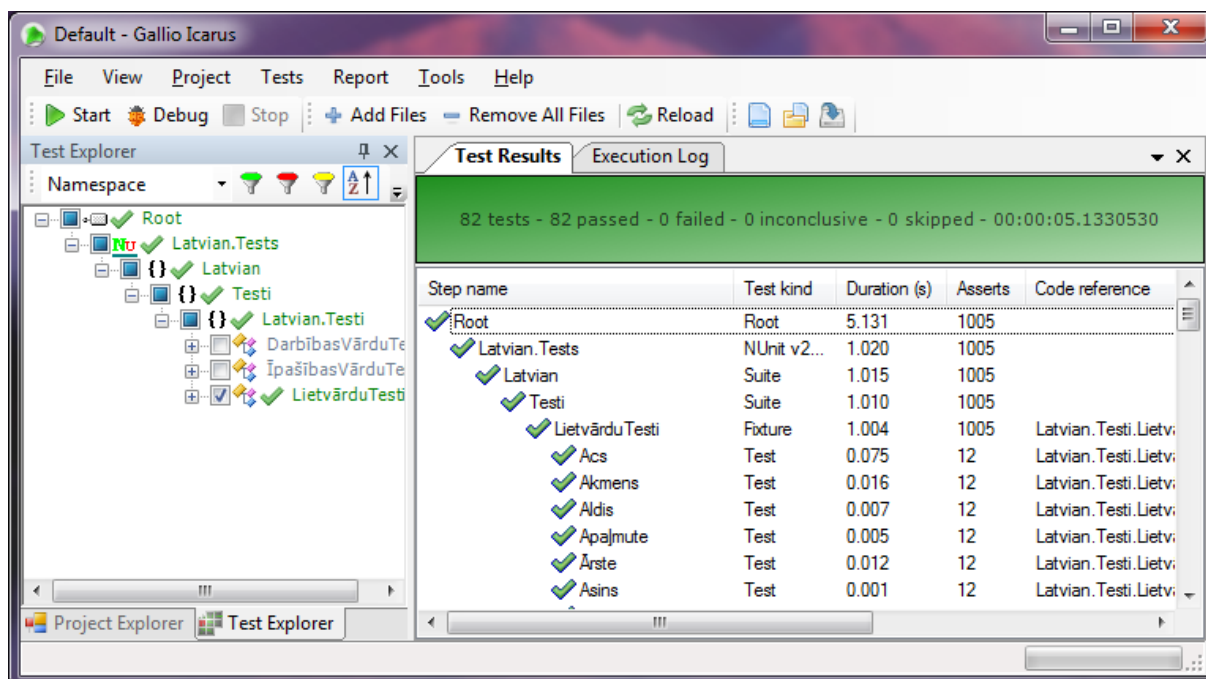
Visi testi izpildās pareizi.



3.2. Lietvārdu testi

Lai gan lielākā daļa lietvārdu ir regulāri lokāmi, ir vairāki izņēmumi. Katrai deklinācijai tika izlocīti pāris regulāri lokāmi lietvārdi, tika iztestētas visas līdzskaņu mijas, kā arī pārbaudīti visi gramatikas rokasgrāmatās uzskaitītie izņēmumi. Tika pārbaudīta arī atgriezenisko un nelokāmo lietvārdu locīšana.

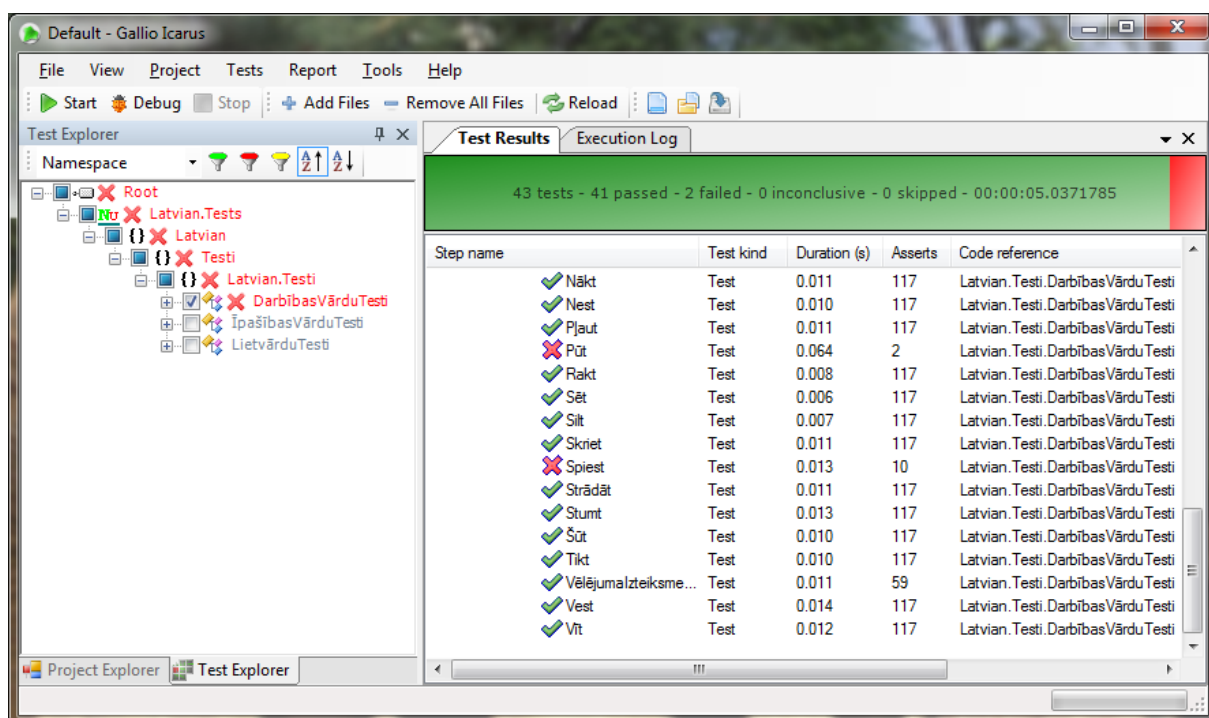
Visi testi izpildās pareizi.



3.3. Darbības vārdu testi

Latviešu valodā ir tikai trīs nekārtnie darbības vārdi, kas tiek locīti īpatnēji. Pārējiem darbības vārdiem būtu jālokās pēc aprakstītajiem likumiem. Vairāki darbības vārdi tika notestēti, izlokot tos visos iespējamajos locījumos. Citi darbības vārdi tika notestēti pa izteiksmēm. Piemēram, īstenības izteiksmē notestēti viens vai vairāki katras konjugācijas, katras grupas darbības vārdi gan ar, gan bez mijām. Līdz ar to patiesais notestētais darbības vārdu skaits ir lielāks nekā redzams testu rezultātos.

Ne visi testi izpildās.



3.3.1. Darbības vārda „pūst” problēmu ziņojums

Darbības vārds *pūst* tiek nepareizi izlocīts darāmās kārtas īstenības izteiksmes vienkāršās tagadnes vienskaitļa otrajā personā. Atbildei būtu jābūt „pūt” (*tu pūt*), taču tā vietā tiek izlocīts kā „pūs” (*tu pūs*).

Darbības vārda forma tiek veidota no tagadnes celma, kas ir *pūš* (es *pūšu*). Šis vārds ir I konjugācijas 4. grupas tiešs darbības vārds, tāpēc galotne -i šajā personā un skaitlī nepienākas. Dažiem I konjugācijas darbības vārdiem šajā personā un skaitlī ir līdzskaņu mija. Tā kā locītājam tiek padota vsk. 1. personas forma, kurai jau ir mija, tā tiek atcelta. Piemēram, ir mija *s* → *š* (es *dzēšu*, tu *dzēs*). Tā kā tiek padota forma „dzēšu”, tad notiek mijas atcelšana: *š* → *s*. Tāpēc *š* tiek pārvērsts par *s*: es *pūšu* → tu *pūs*. Nekādas papildus darbības pēc prasībām nav jāveic.

3.3.2. Darbības vārda „spiest” problēmu ziņojums

Darbības vārds *spiest* tiek nepareizi izlocīts darāmās kārtas īstenības izteiksmes vienkāršās tagadnes vienskaitļa otrajā personā. Atbildei būtu jābūt „spied” (*tu spied*), taču tā vietā tiek izlocīts kā „spiez” (*tu spiez*).

Tāpat kā *pūst*, *spiest* ir I konjugācijas 4. grupas darbības vārds. Arī šeit ir atcelta līdzskaņu mija *ž* → *z* (es *spiežu* → tu *spiez*).

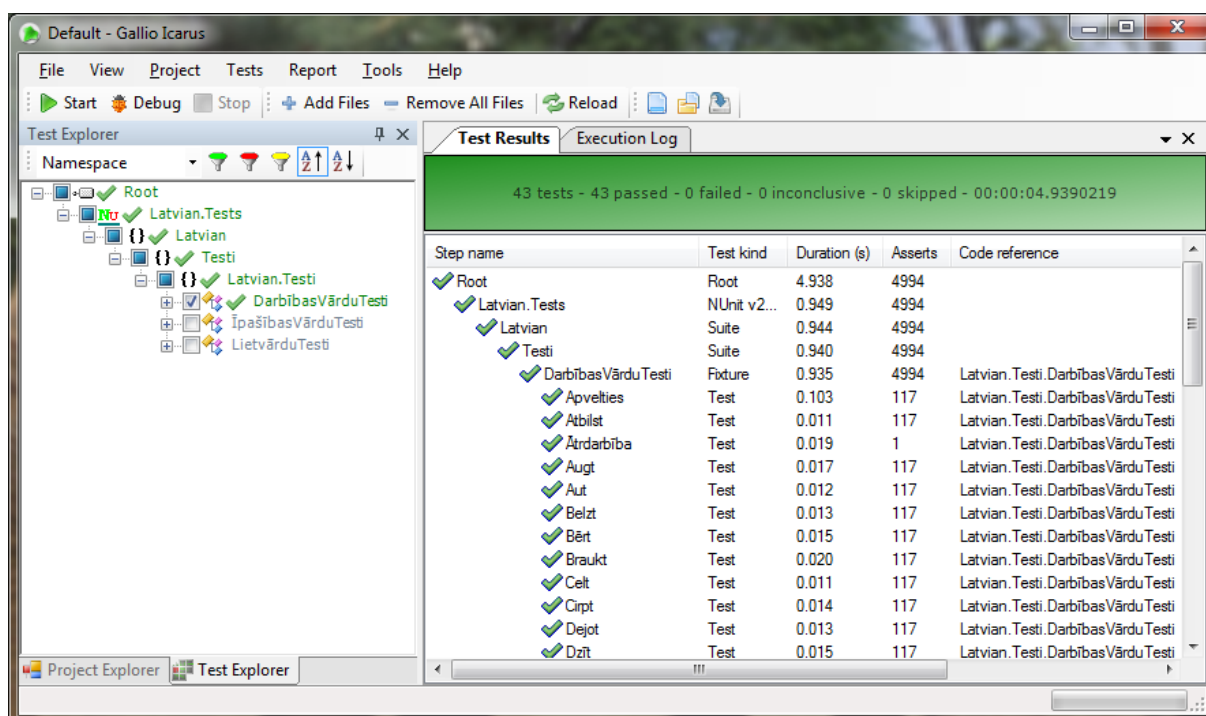
3.3.3. Darbības vārdu „pūst” un „spiest” problēmu novēršana

Acīmredzot vārdi netiek pareizi izlocīti, jo līdzskaņu mijas nav aprakstītas pilnīgi.

Taču, pārskatot I konjugācijas 4. grupas vārdu celmus, tika pamanīts, ka pagātnes celma pēdējais līdzskanis sakrīt ar līdzskani personas vienskaitļa otrajā formā. Autors izvirzīja hipotēzi, ka I konjugācijas 4. grupas vārdiem, kam ir līdzskaņu mija darāmās kārtas īstenības izteiksmes vienkāršās tagadnes vienskaitļa personas otrajā formā, līdzskaņu mija notiek nevis uz prasībās aprakstītajām mijām, bet gan uz pagātnes celma pēdējo līdzskani.

Tika veikta izmaiņa locītājā un tika palaisti visi testi. Visi iepriekš testētie I konjugācijas 4. grupas darbības vārdi joprojām tika izlocīti pareizi un gan pūst, gan spiest arī tika izlocīti pareizi.

Pēc veiktās izmaiņas visi darbības vārdu testi izpildās pareizi.



3.4. Divdabju testi

Divdabji netika testēti, jo tie tiek locīti tāpat kā īpašības vārdi.

3.5. Veikspējas testi

Vienībtestos katrai vārdšķirai tika uzņemts laiks, cik ilgi notiek locīšana. Rezultāti iekļaujas izvirzītajās nefunkcionālajās veikspējas prasībās, jo testi izpildās.

4. Programmatūras pirmkoda fragmenti

Viss programmatūras pirmkods ir par lielu, lai to iekļautu šajā darbā, tāpēc iekļauti tikai daži tā fragmenti:

- ģpašības vārdu locīšanas klase,
- otrās deklinācijas lietvārdu locīšanas klase,
- darbības vārdu locīšanas klase,
- darāmās kārtas tagadnes divdabja locīšanas klase,
- divi vienībtesti.

4.1. ĢpašībasVārds

```
using System;
```

```
namespace Latvian.Vārdšķiras.ĢpašībasVārdi
```

```
{  
    /// <summary>  
    /// Ģpašības vārdu leksiskā nozīme nemainās no tā, vai ģpašības vārdam ir noteiktā vai  
    /// nenoteiktā galotne. Šīs galotnes rāda priekšmetu pazīmi kā zināmu vai nezināmu.  
    /// <see>http://www.liis.lv/latval/morfol/ipv-gal.htm</see>  
    /// </summary>  
    public enum Galotne  
    {  
        Nenoteiktā,  
        Noteiktā  
    }  
  
    // lai gan sakrīt ar lietvārdu locījumiem,  
    // ģpašības vārdu locījumi ir atšķirīgi  
    // (lietvārdiem ir vokatīvs)  
    public enum Locījums  
    {  
        Nominatīvs,  
        Ģenitīvs,  
        Datīvs,  
        Akuzatīvs,  
        Lokatīvs  
    }  
  
    /// <summary>  
    /// Ģpašības vārdi ir patstāvīgi, lokāmi vārdi, kas nosauc priekšmetu vai dzīvu būtnu pazīmes.  
    /// <see>http://www.liis.lv/latval/morfol/ipasibv.htm</see>  
    /// </summary>  
    public class ĢpašībasVārds : Vārds  
    {  
        /// <summary>  
        /// Izveido jaunu ģpašības vārdu no normālformas.  
        /// </summary>  
        /// <param name="vārds">Ģpašības vārds pamata pakāpes vienskaitļa nominatīvā ar vīriešu  
        /// dzimtes nenoteikto galotni.</param>  
        public ĢpašībasVārds(string vārds)  
        {  
            if (vārds == null)  
                throw new ArgumentNullException("vārds");  
            if (!vārds.BeidzasAr("s", "š"))  
                throw new ArgumentException("Ģpašības vārdam jābūt pamata pakāpes vienskaitļa nominatīvā ar  
                vīriešu dzimtes nenoteikto galotni");  
        }  
    }  
}
```

```

        Galotne = vārds.BeidzasAr("s") ? "s" : "š";
        Celms = vārds.AtmetGalotni(Galotne);
    }

    // ...

    public virtual string NominativaGalotne(Dzimte dzimte, Skaitlis skaitlis, Galotne galotne)
    {
        if (skaitlis == Skaitlis.Vienskaitlis)
        {
            if (dzimte == Dzimte.Vīriešu)
            {
                return (galotne == ĪpašībasVārds.Galotne.Nenoteiktā) ? Galotne : "ais";
            }
            else
            {
                return (galotne == ĪpašībasVārds.Galotne.Nenoteiktā) ? "a" : "ā";
            }
        }
        else
        {
            if (dzimte == Dzimte.Vīriešu)
            {
                return (galotne == ĪpašībasVārds.Galotne.Nenoteiktā) ? "i" : "ie";
            }
            else
            {
                return (galotne == ĪpašībasVārds.Galotne.Nenoteiktā) ? "as" : "ās";
            }
        }
    }

    // ...

    public string Nominatīvs(Dzimte dzimte = Dzimte.Vīriešu, Skaitlis skaitlis = Skaitlis.Vienskaitlis,
        Galotne galotne = ĪpašībasVārds.Galotne.Nenoteiktā)
    {
        return Celms + NominativaGalotne(dzimte, skaitlis, galotne);
    }

    #region Salīdzināmās pakāpes
    /// <summary>
    /// Īpašības vārds pamata pakāpē nosauc īpašību kā neatkarīgu, patstāvīgu.
    /// </summary>
    public virtual ĪpašībasVārds PamataPakāpe
    {
        get { return new ĪpašībasVārds(Celms.Apcirpt("vis", paKreisi: true, paLabī: false)
            .Apcirpt("āk", paKreisi: false, paLabī: true), Galotne); }
    }

    /// <summary>
    /// Īpašības vārds pārākajā pakāpē norāda, ka īpašība priekšmetam vai parādībai piemīt
    /// lielākā vai mazākā mērā nekā kādam citam līdzīgam priekšmetam vai parādībai.
    /// </summary>
    public virtual ĪpašībasVārds PārākāPakāpe
    {
        // Pārākās pakāpes formu darina no īpašības vārda celma, pievienojot tam piedēkli -āk-.
        get { return new ĪpašībasVārds(PamataPakāpe.Celms + "āk", Galotne); }
    }
    #endregion
}

```

4.2. Otrās Deklinācijas Lietvārds

```
/// <summary>
/// II deklinācija.
/// Pie II deklinācijas pieder vīriešu dzimtes lietvārdi, kam vienskaitļa nominatīvā ir galotne -is,
/// kā arī izņēmumi: mēness, akmens, asmens, rudens, ūdens, zibens, suns, sāls.
/// </summary>
public class OtrāsDeklinācijasLietvārds : DeklinējamsLietvārds
{
    public OtrāsDeklinācijasLietvārds(string vārds)
    {
        if (vārds == null)
            throw new ArgumentNullException("vārds");
        if (!vārds.BeidzasAr("s"))
            throw new ArgumentException("Lietvārdam jābūt vienskaitļa nominatīvā");

        Galotne = vārds.BeidzasAr("is") ? "is" : "s";
        Celms = vārds.AtmetGalotni(Galotne);
    }

    public override Deklinācija Deklinācija
    {
        get { return Deklinācija.II; }
    }

    public override Dzimte Dzimte
    {
        get { return Dzimte.Vīriešu; }
    }

    public override string NominatīvaGalotne(Skaitlis skaitlis)
    {
        return (skaitlis == Skaitlis.Vienskaitlis) ? Galotne /* -is, -s */ : "i";
    }

    public override string ĢenitīvaGalotne(Skaitlis skaitlis)
    {
        // lai gan suns ir izņēmums ar -s galotni, vienskaitlī tas neuzvedas kā izņēmums
        return (skaitlis == Skaitlis.Vienskaitlis) ?
            (Galotne == "s" && !Celms.BeidzasAr("sun") ? "s" : "a") : "u";
    }

    // ...

    protected override string LīdzskaņuMija(Locījums locījums, Skaitlis skaitlis)
    {
        // II deklinācijas lietvārdiem līdzskaņu mija raksturīga
        // vienskaitļa ģenitīvā un visos daudzskaitļa locījumos.
        if (locījums == Locījums.Ģenitīvs || skaitlis == Skaitlis.Daudzskaitlis)
        {
            // Līdzskaņu mijas nav:
            bool bezMijas = (Celms + "is").BeidzasAr(
                "tētis", "viesis", "astis", "skatis",
                "jis", "ķis", "ģis", "ris");

            bool suns = skaitlis == Skaitlis.Vienskaitlis && (Celms + "s").BeidzasAr("suns");

            // Lietvārdiem līdzskaņu mija ir tikai daudzskaitlī.
            // (tātad nav vienskaitlī)
            bool izņēmumi = skaitlis == Skaitlis.Vienskaitlis && (Celms + "s").BeidzasAr(
                "akmens", "asmens", "rudens", "zibens", "ūdens", "mēness", "sāls");

            if (suns || (!bezMijas && !izņēmumi))
            {
                string arMiju = Celms;

                arMiju = arMiju.LīdzskaņuMija("sn", "šņ")
                    .LīdzskaņuMija("zn", "žņ")
                    // ...
                    .LīdzskaņuMija("n", "ņ")
                    .LīdzskaņuMija("l", "ļ");

                return arMiju;
            }
        }
    }
}
```

4.3. DarbībasVārds

```
public class DarbībasVārds : Vārds
{
    private string nenoteiksme;
    private string nenoteiksmesCelms;
    private string tagadnesCelms;
    private string pagātnesCelms;
    private Konjugācija konjugācija;
    private int grupa;

    public DarbībasVārds(string nenoteiksme, string tagadne, string pagātne,
        Konjugācija konjugācija, int grupa)
    {
        this.nenoteiksme = nenoteiksme;
        this.konjugācija = konjugācija;
        this.grupa = grupa;

        NenoteiksmesCelms = nenoteiksme;
        TagadnesCelms = tagadne;
        PagātnesCelms = pagātne;
    }

    public string Nenoteiksme
    {
        get { return nenoteiksme; }
    }

    public string NenoteiksmesCelms
    {
        get { return nenoteiksmesCelms; }
        protected set
        {
            if (!value.BeidzasAr("ties", "t"))
                throw new ArgumentException();

            if (value.BeidzasAr("ties")) {
                nenoteiksmesCelms = value.AtmetGalotni("ties");
            } else {
                nenoteiksmesCelms = value.AtmetGalotni("t");
            }
        }
    }

    public string ĪstenībasIzteiksmesVienkāršāPagātne(PersonasForma persona)
    {
        string vārds = PagātnesCelms;

        switch (persona)
        {
            // Vienkāršajā pagātnē vienskaitļa 1.personas formā darbības vārdiem ir galotne -u vai -os.
            // II konjugācijas darbības vārdiem sakrīt vienkāršās pagātnes un vienkāršās tagadnes viensk
            aitļa 1.personas formas.
            case PersonasForma.VienskaitļaPirmāPersona:
                vārds += Tiešs ? "u" : "os";
                break;

            // Vienkāršajā pagātnē vienskaitļa 2.personas formā darbības vārdiem ir galotne -i vai -ies.
            case PersonasForma.VienskaitļaOtrāPersona:
                vārds += Tiešs ? "i" : "ies";
                break;

            // Vienkāršajā pagātnē daudzskaitļa 1.personas formā visiem darbības vārdiem ir galotne -
            ām vai -āties.
            case PersonasForma.DaudzskaitļaPirmāPersona:
                vārds += Tiešs ? "ām" : "āties";
                break;

            // Vienkāršajā pagātnē daudzskaitļa 2.personas formā visiem darbības vārdiem ir galotne -
            āt vai -āties.
            case PersonasForma.DaudzskaitļaOtrāPersona:
                vārds += Tiešs ? "āt" : "āties";
                break;
        }
    }
}
```

```

        // Vienkāršajā pagātnē 3.personas formā visiem darbības vārdiem ir galotne -a vai -ās.
        case PersonasForma.VienskaitļaTrešāPersona:
        case PersonasForma.DaudzskaitļaTrešāPersona:
            vārds += Tiešs ? "a" : "ās";
            break;
    }
}

```

4.4. DarāmāsKārtasTagadnesDivdabis

```

using Latvian.Vārdšķiras.ĪpašībasVārdi;

namespace Latvian.Vārdšķiras.DarbībasVārdi.Divdabji
{
    public class DarāmāsKārtasTagadnesDivdabis : ĪpašībasVārds
    {
        public DarāmāsKārtasTagadnesDivdabis(DarbībasVārds vārds) : base(
            (vārds.TagadnesCelms.BeidzasAr("š", "ž") ? vārds.PagātnesCelms : vārds.TagadnesCelms)
            + "ošs")
        {
        }
    }
}

```

4.5. Lietvārdu vienībtests

```

Lietvārds māja = new CeturtāsDeklinācijasLietvārds("māja");
Assert.AreEqual(Dzimte.Sieviešu, māja.Dzimte);
Assert.AreEqual(Deklinācija.IV, (māja as DeklinējamsLietvārds).Deklinācija);
Assert.AreEqual("māja", māja.Nominatīvs(Skaitlis.Vienskaitlis));
Assert.AreEqual("mājas", māja.Nominatīvs(Skaitlis.Daudzskaitlis));
Assert.AreEqual("mājas", māja.Ģenitīvs(Skaitlis.Vienskaitlis));
Assert.AreEqual("māju", māja.Ģenitīvs(Skaitlis.Daudzskaitlis));
Assert.AreEqual("mājai", māja.Datīvs(Skaitlis.Vienskaitlis));
Assert.AreEqual("mājām", māja.Datīvs(Skaitlis.Daudzskaitlis));
Assert.AreEqual("māju", māja.Akuzatīvs(Skaitlis.Vienskaitlis));
Assert.AreEqual("mājas", māja.Akuzatīvs(Skaitlis.Daudzskaitlis));
Assert.AreEqual("mājā", māja.Lokatīvs(Skaitlis.Vienskaitlis));
Assert.AreEqual("mājās", māja.Lokatīvs(Skaitlis.Daudzskaitlis));

```

4.6. Īpašības vārdu pakāpju vienībtests

```

[Test]
public void PakāpjuCelmi()
{
    ĪpašībasVārds labs;
    labs = new ĪpašībasVārds("labs");
    Assert.AreEqual("lab", labs.PamataPakāpe.Celms);
    Assert.AreEqual("labāk", labs.PārākāPakāpe.Celms);
    Assert.AreEqual("vislabāk", labs.VispārākāPakāpe.Celms);
    labs = labs.PārākāPakāpe;
    Assert.AreEqual("lab", labs.PamataPakāpe.Celms);
    Assert.AreEqual("labāk", labs.PārākāPakāpe.Celms);
    Assert.AreEqual("vislabāk", labs.VispārākāPakāpe.Celms);
    labs = labs.VispārākāPakāpe;
    Assert.AreEqual("lab", labs.PamataPakāpe.Celms);
    Assert.AreEqual("labāk", labs.PārākāPakāpe.Celms);
    Assert.AreEqual("vislabāk", labs.VispārākāPakāpe.Celms);
}

```

5. Projekta organizācija

Latviešu valodas vārdu locīšanas bibliotēka tika izstādāta, par pamatu ņemot spirālveida un prototipa modeli.

Vispirms tika veikta viena teorētiskā materiāla – latviešu valodas morfoloģijas apraksts, kas kalpoja par programmatūras prasību specifiku, izpēte un lietvārdu locītāja prototipa izveide. Prototipa dizains tika atzīts par labu un paplašināmu un līdzīgi tika izveidoti locītāji pārējām vārdšķirām (īpašības vārdi, pēc tam darbības vārdi un arī divdabji). Tika izveidoti vienībtesti, lai pārbaudītu locīšanas pareizību. Izstrādes laikā vairākas reizes tika veikta restrukturizācija pēc jaunu programmēšanas valodas iespēju atklāšanas un papildus teorētiskā materiāla izpētes, kā arī tika veikta papildu vienībtestu pievienošana un testēšana. Pēc jaunu avotu apguves, tika papildināti vienībtesti, pielabots locītājs un beidzot atjauninātas prasības. Visbeidzot tika veikta produkta dokumentācijas galīgās versijas izstrāde, kur programmatūras prasību specifikācijā tika apkopoti un tabulēti materiāli no vairākiem avotiem, un aprakstīts projektējums.

Nepieciešamais teorētiskais materiāls tika ņemts no latviešu valodas rokasgrāmatām un projekta izstrāde tika veikta patstāvīgi, tajā piedaloties vienam cilvēkam (darba autoram), kas veica projektētāja, programmētāja, dokumentācijas izstrādātāja un testētāja funkcijas. Darba vadītājs palīdzēja testēšanā, piedāvājot papildus testēšanas piemērus.

Projekta izstrāde notika, izmantojot Microsoft Visual Studio izstrādes vidi, .NET Framework programmēšanas ietvaru, C# programmēšanas valodu un git konfigurācijas pārvaldības rīku. Kvalifikācijas darbs un projekta pavadošā dokumentācija tika veidota, izmantojot Microsoft Word.

6. Konfigurācijas pārvaldība

Programmatūras produkta konfigurācijas pārvaldībai tika izmantota git versiju pārvaldības sistēma. Šīs sistēmas izvēli noteica

- autora iepriekšējā pieredze ar to,
- darbu ar šo sistēmu veica tikai viens cilvēks,
- nav nepieciešams centralizēts repozitorijs,
- ļoti noderīgā iespēja viegli veidot un apvienot vairākus atzarus jaunu iespēju un restrukturizācijas izmēģināšanai.

Projekta kods kopā ar konfigurācijas repozitoriju tika glabāts uz izstrādes datora cietā diska, katras dienas beigās, kad tika veiktas izmaiņas, tika izveidota kopija uz ārējā cietā diska un attālināta kopija mākonī.

Dokumentācijas konfigurācijas pārvaldībai tika izmantota teksta redaktora iebūvētā iespēja „Track Changes”.

7. Kvalitātes nodrošināšana

Lai nodrošinātu kvalitatīva produkta izstrādi tika veiktas šādas darbības:

- projektu pavadošā dokumentācija tika izstrādāta, vadoties pēc atbilstošajiem Latvijas Valsts standartiem,
- bibliotēkas kodā ir komentēta gandrīz katra metode (izņemot tās, uz kurām attiecināmi vieni un tie paši komentāri),
- bibliotēkas kodā ir ievietoti prasību specifikācijas fragmenti, lai, kodu lasot, nebūtu katreiz jālieto programmatūras prasību specifikācija,
- gan komentāri, gan klašu/mainīgo/failu nosaukumi, gan vienībtesti ir latviešu valodā ar visiem latviešu burtiem, jo to atļauj izvēlētā programmēšanas valoda [6], kas palīdz vieglāk uztvert kodu, jo nav jāmeklē atbilstošie termini citā valodā,
- pēc vienībtestiem var pilnībā izsecināt, kā lietojama bibliotēka,
- katrai produkta daļai tikai veikta vienībtestēšana, lai pārliecinātos par tās pareizu darbību, kā arī katrai vārdšķīrai tika izveidoti vienībtesti katram rokasgrāmatā atrodamajam izņēmumam.

8. Darbietelpības novērtējums

Visvairāk laika tika veltīts programmatūras prasību apkopošanai, vienībtestu izstrādei un restrukturizācijai, lai katru reizi censtos iegūt bez iespējas ērtāk lietojamu bibliotēkas saskarni.

Bibliotēkas pirmkods sastāv no vairāk kā 2000 koda rindiņām (tā kā komentāri ļoti atvieglo koda lasīšanu un var aizstāt prasību specifikāciju, tie tika ieskaitīti) un vienībtesti kopumā aizņem aptuveni 8000 koda rindiņas. Lai gan pašas bibliotēkas kopējais koda rindiņu skaits nav pārāk liels, uzsvars šajā projektā tika likts uz kvalitāti nevis kvantitāti, vairākas reizes pārdomājot un pārveidojot bibliotēkas saskarni un lietojamību.

8.1. Darbietelpības prognozēšana, izmantojot COCOMO Basic

Rādītāji mazam projektam [7]:

$$a_b = 2,4, b_b = 2,5, c_b = 1,05, d_b = 0.38$$

Kvalifikācijas darba rādītāji:

$$KLOC = 2 \text{ (koda rindiņu skaits tūkstošos)}$$

$$KLOC = 10 \text{ (koda, ieskaitot vienībtestus, rindiņu skaits tūkstošos)}$$

Izstrādes laiks

$$D = c_b(a_b(KLOC)^{b_b})^{d_b} = 2.82910 \approx 3 \text{ mēneši bez vienībtestiem}$$

$$D = c_b(a_b(KLOC)^{b_b})^{d_b} = 13.0518 \approx 13 \text{ mēneši ar vienībtestiem}$$

9. Rezultāti

Darba izstrādes gaitā ir izveidota funkcionējoša latviešu valodas vārdu locīšanas bibliotēka, kas spēj izlocīt visus regulāros lietvārdus, īpašības vārdus un darbības vārdus, kā arī divdabjus, atpazīst lielu daļu izņēmumu un piedāvā paraugus neregulāru vārdu locīšanai. Darbs rezultāts atbilst izstrādātajai prasību specifikācijai. Tika veikta visas izstrādes dokumentēšana un vienībtestēšana.

Darba izpildes gaitā autors apguva jaunas iemaņas programmēšanas valodā C#, paplašināja savu vārdu krājumu, padziļināja savas zināšanas savas dzimtās valodas gramatikā un uzlaboja pareizrakstības iemaņas.

10. Secinājumi

Pēc darba paveikšanas autors secināja, ka tradicionālie izstrādes modeļi kā, piemēram, ūdenskrituma modelis nebūtu piemēroti šādas programmatūras izstrādei, jo, lai arī lielākā daļa latviešu valodas gramatikas likumu ir aprakstīti, testēšanas laikā tika atrastas neaprašītas situācijas, kuru rezultātā nācās pamainīt prasību specifikāciju. Vispiemērotākais modelis būtu bijis tāds, kur sāk ar vienībtestu veidošanu (test driven development).

Tika izdarīti arī vairāki secinājumi par latviešu valodu. Pēc autora domām latviešu valoda gramatikas ziņā ir viena no vienkāršākajām dabīgajām indoeiropiešu valodām, jo vārdu locīšana ir ļoti regulāra (par ko liecina fakts, ka ir iespējams izstrādāt automatizētu datorprogrammu, kas to spēj darīt) un ir tikai saujiņa izņēmumu, kur liela daļa ir ikdienā mazlietoti vārdi. Tika novērots, ka latviešu valodai ir arī tendences kļūt aizvien regulārākai, piemēram, lielākā daļa aizgūto darbības vārdu kļūst par II konjugācijas darbības vārdiem (piemēram, importēt, eksportēt, sērfot), kuri ir visvieglāk izlokāmi. Latviešu valodā darbības vārdu locīšana pagātnē ir daudz vienkāršāka nekā tagadnē un latviešu valodas apguvējiem ieteiktu to apgūt vispirms.

Vēl tika nonākts pie secinājuma, ka objektorientētā programmēšana bija pats labākais risinājums aprakstīt latviešu valodas vārdus un locījumaus. Lai gan tas netika uzskaitīts, projektā tika izmantoti gandrīz visi objektorientētās programmēšanas principi.

Pateicības

Izsaku pateicību darba vadītājam Normundam Grūzītim par atsaucību, atsaucsmēm un palīdzību testēšanā, kā arī Agrim Šostakam par darba vadītāja ieteikšanu.

Paldies arī Madaram Virzam, kura kvalifikācijas darbu ņemu par paraugu šī darba noformēšanai.

Paldies par iedvesmu Kasparam Foigtam, kurš pirmais izveidoja tiešsaistes lietvārdu locītāju tālajā 2003. gadā.

Izmantotā literatūra un avoti

[1] *MORFOLOĢIJA* [tiešsaiste]. [atsauce 24.05.2011]. Pieejams:

<http://www.liis.lv/latval/morfol/index.htm>

[2] **B.Ceplīte, L.Ceplītis.** *Latviešu valodas praktiskā gramatika*. Rīga „Zvaigzne”, 1991. 239 pp.

[3] *OpenOffice 2.x.x Latviešu valodas pareizrakstības un pārvešanas moduļu uzstādīšana*

[tiešsaiste]. [atsauce 24.05.2011]. Pieejams: [http://openoffice-](http://openoffice-lv.sourceforge.net/openoffice.html)

[lv.sourceforge.net/openoffice.html](http://openoffice-lv.sourceforge.net/openoffice.html)

[4] **Pēteris Paikens.** *Latviešu valodas automatizēta morfoloģiskā un sintaktiskā analīze*. Rīga, 2007. 58 pp.

[5] **Mārcis Pinnis.** *Uzlabota latviešu valodas analīzes un sintēzes sistēma*. Rīga, 2007. 63 pp.

[6] *C# Language Specification. Standard ECMA-334. 4th Edition / June 2006* [tiešsaiste].

[atsauce 24.05.2011]. Pieejams: <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>

[7] *Basic COCOMO - Wikipedia, the free encyclopedia* [tiešsaiste]. [atsauce 24.05.2011].

Pieejams: http://en.wikipedia.org/wiki/COCOMO#Basic_COCOMO

[8] *Decorator Design Pattern in C#* [tiešsaiste]. [atsauce 24.05.2011]. Pieejams

<http://www.dofactory.com/Patterns/PatternDecorator.aspx>

[9] *Abstract Factory Design Pattern in C#* [tiešsaiste]. [atsauce 24.05.2011]. Pieejams:

<http://www.dofactory.com/Patterns/PatternAbstract.aspx>

[10] *NUnit testing framework for .NET* [tiešsaiste]. [atsauce 24.05.2011]. Pieejams:

<http://www.nunit.org/>

[11] *Gallio - Automation Platform for .NET* [tiešsaiste]. [atsauce 24.05.2011]. Pieejams:

<http://www.gallio.org/Docs.aspx>

Kvalifikācijas darbs „*Latviešu valodas vārdu locītājs*” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Pēteris Nikiforovs* _____ .05.2011.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *Dr.sc.comp., Normunds Grūzītis* _____ .05.2011.

Recenzents: *M.soc.zin., Māris Dreimanis*

Darbs iesniegts 26.05.2011.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Dainis Dosbergs* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2011. prot. Nr. _____, vērtējums _____ (_____)

Komisijas sekretārs(-e): _____