**VIDZEMES AUGSTSKOLA**

Inžinierzinātņu fakultāte

Studiju programma: Informācijas tehnoloģijas

2. kurss

**Programmas izstrāde C++ vidē**

Teksta redaktors ‘SimplePad’

Izstrādāja: Aksels Skvorcovs

Pieņēma: A. Fjodorovs

2019

# Anotācija

Kursa darba pārskatu veidu 5 nodaļas tai skaitā bibliogrāfiskais saraksts un vienpadsmit attēli.

Pārskatā ir dots lietojumprogrammas “SimplePad” apraksts, aplūkotas klases, kas izmantotas programmā, kā arī, izmantotās bibliotēkas apraksts un izmantošanas piemērs, programmatūras testa piemērs un demonstrācijas.

Programma sastādīta programmēšanas valodā C++ izmantojot Qt Creator IDE.

# Saturs

[**Anotācija**](#_qz3cly2ieuha) **2**

[**Saturs**](#_bomkank5zfwl) **3**

[**1. Uzdevumu nostādne**](#_cik3so4dqj9f) **4**

[1.1 Teksta redaktori](#_q9gcw7d8tyb7) 4

[**2. Programmas apraksts**](#_s1mb4nvz8bf4) **5**

[2.1 Main.cpp](#_8oqqchsxgwq0) 5

[2.1.1 Objekti un funkcijas](#_xrur38zevsy4) 5

[2.2 simplepad.cpp](#_tdmspixw0jme) 5

[2.2 Funkcijas](#_sty43k6f304x) 6

[**3. Bibliotēka ‘Qt’**](#_dcsvjnwnz3qx) **7**

[3.1 Izmantotās bibliotēkas apraksts](#_40kv9elyc8s4) 7

[3.2 Izmantotās bibliotēkas pielietošana programmā](#_f5eb96uh2260) 7

[**4. Test piemēri**](#_rdtafkbaxbs3) **8**

[4.1 Programmas darbības demonstrācija](#_a2ruo8c8z0j9) 8

[4.1.1 Programmas palaišaina](#_kr9sxwa1tnc3) 8

[4.1.2 Jauns fails](#_9g8osxyrztps) 9

[4.1.3 Teksta apstrāde](#_7avybedvxkkd) 9

[4.1.4 Teksta kopēšana un ielīmēšana](#_2aa14y3ja3li) 10

[4.1.5 Informācijas logs](#_cyn82qhjwzge) 10

[4.1.6 Saglabāt failu](#_hdvn5bl4oqi9) 11

[4.1.7 Printēt failu](#_mwh1medpsley) 11

[4.1.8 Faila atvēršana](#_g5iqsjh03gb8) 12

[**5. Bibliogrāfiskais saraksts.**](#_i0nsctstdnmu) **13**

# 1. Uzdevumu nostādne

Uzdevuma nostādne ir izstrādāt teksta redaktora lietojumprogrammu, kas spēj atvērt, rediģēt un saglabāt vienkāršus teksta failus.

## 1.1 Teksta redaktori

Teksta redaktors ir datorprogramma, kas ļauj sastādīt un aplūkot parastus teksta failus.

Tipiskas teksta redaktoru tehniskās iespējas ir teksta kopēšana, izgriešana un ielīmēšana, kā arī simbolu virkņu meklēšana un aizvietošana.

Teksta redaktors ir viena no programmām, kas parasti tiek iekļauta jebkurā operētājsistēmu distributīvā, lai būtu iespēja rediģēt konfigurācijas un citus failus.

Tekstprocesorus, piemēram, OpenOffice.org Writer, AbiWord, Microsoft Word, nepieskaita pie teksta redaktoriem, jo šajās programmās veidotie dokumenti nav parasti teksta faili un nav izmantojami ar parastiem teksta redaktoriem.

# 

# 

# 2. Programmas apraksts

Programmas struktūra ietver divas klases: main.cpp un simplepad.cpp

## 2.1 Main.cpp

Main.cpp ir galvenā klase, jo tā satur main funkciju, kas nodrošina programmas pamatdarbību. Tajā tiek izveidots objekts no klases *QApplicaton*. Šis objekts ir svarīgs, jo tas tiek galā ar ievades argumentiem un izveido notikumu ciklu (event loop). Notikumu cikls ir cikls, kurš gaida lietotāja ievadi grafiskajā lietotāja saskarnē. *SimplePad* objekts nodrošina lietotāja grafiskās saskarnes izveidošanu un attēlošanu ar metodi *.show()*, kura padara lietotāja grafisko saskarni redzamu. Metode *.exec()* inicializē un uzsāk notikumu ciklu, kurš gaida lietotāja ievadi grafiskajā lietotāja saskarnē.

### 2.1.1 Objekti un funkcijas

Objekti: QApplication, SimplePad

Funkcijas: .show(), .exec()

## 2.2 simplepad.cpp

Simplepad.cpp klase nodrošina grafiskās lietotāja saskarnes izkārtojuma funkcionalitāti, tas ir - nospiežot, piemēram, pogu, tiks izpildīta funkcija, piemēram, saglabāt failu. Simplepad.cpp satur construktoru, destruktoru un 11 funkcijas.

### 

### 2.2 Funkcijas

Funkcijas:

on\_New\_triggered() - Notīra teksta laukumu un sagatavoto jaunai ievadei.

on\_Open\_triggered() - Atver logu, kurā ļauj izvēlēties failu. Pēc faila izvēles mēģina to atvērt. Ja neizdodas, tad izvada kļūdas paziņojumu. Ja izdodas, tad faila saturu izvada teksta redaktora rakstīšanas laukumā.

on\_Save\_As\_triggered() - Atver logu, kas ļauj saglabāt failu. Ja saglabājot failu rodas kļūda, tad tiek izvadīts kļūdas paziņojums. Ja izdodas saglabāt, tad failu saglabā.

on\_Exit\_triggered() - Aizver lietojumprogrammu.

on\_Copy\_triggered() - Iezīmēto lauku nokopē.

on\_Paste\_triggered() - Ielīmē iepriekš nokopēto tekstu.

on\_Undo\_triggered() - Atsauc iepriekšējo darbību.

on\_Redo\_triggered() - Izpilda atsaukto darbību.

on\_Print\_triggered() - Atver printēšanas logu. Ja neizdodas savienoties ar printeri, tad atgriež kļūdas paziņojumu. Ja izdodas, tad nosūta uz printeri tekstu.

on\_about\_triggered() - Izvada logu ar informatīvu tekstu par programmu.

on\_Change\_font\_triggered() - Atver logu, kurā ļauj izvēlēties fontu. Pēc jauna fonta izvēles nomaina teksta laukuma teksta fontu.

on\_Change\_text\_color\_triggered() - Atver logu, kurā ļauj izvēlēties teksta krāsu. Pēc jaunas krāsas izvēles nomaina teksta laukuma teksta krāsu.

# 3. Bibliotēka ‘Qt’

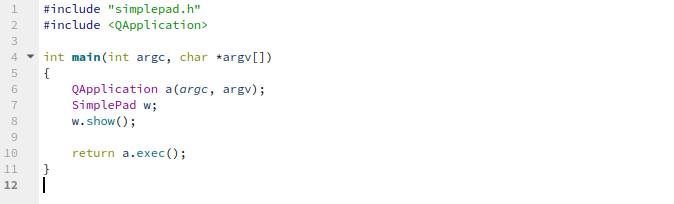
## 3.1 Izmantotās bibliotēkas apraksts

Qt ir bezmaksas un atvērtā koda logrīku rīku komplekts grafisko lietotāja interfeisu izveidei, kā arī starpplatformu lietojumprogrammām, kas darbojas dažādās programmatūras un aparatūras platformās, piemēram, Linux, Windows, MacOS, Android vai iegultās sistēmās ar nelielām izmaiņām vai bez izmaiņām. koda bāze, kamēr tā joprojām ir dzimtā lietojumprogramma ar vietējām iespējām un ātrumu. Qt šobrīd izstrādā uzņēmums Qt, kas ir biržas sarakstā iekļauts uzņēmums, un Qt projekts, kas darbojas atklātā pirmkoda pārvaldībā, iesaistot individuālus izstrādātājus un organizācijas, kas darbojas, lai veicinātu Qt. Qt ir pieejams gan komerciālajās licencēs, gan atvērtā koda GPL 2.0, GPL 3.0 un LGPL 3.0 licencēs.

## 3.2 Izmantotās bibliotēkas pielietošana programmā

Šī bibliotēka ir visas šīs programmas pamatā. Šajā programmā viss balstās uz QApplication piedāvātajām apakšklasēm un metodēm, kas ļauj izveidot funkcionālu grafisko lietotāja saskarni.

Piemērā parādīta main.cpp klase, kurā tiek izveidots QApplication objekts, un pēc tam tiek izpildīta tā funkcija .exec(), kura sāk notikumu ciklu, kurš klausās un gaida lietotāja ievadi.

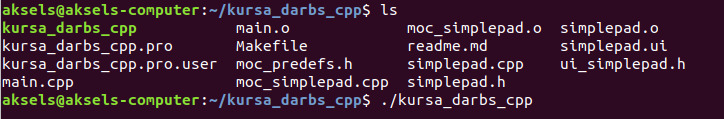
(1. attēls)

# 4. Test piemēri

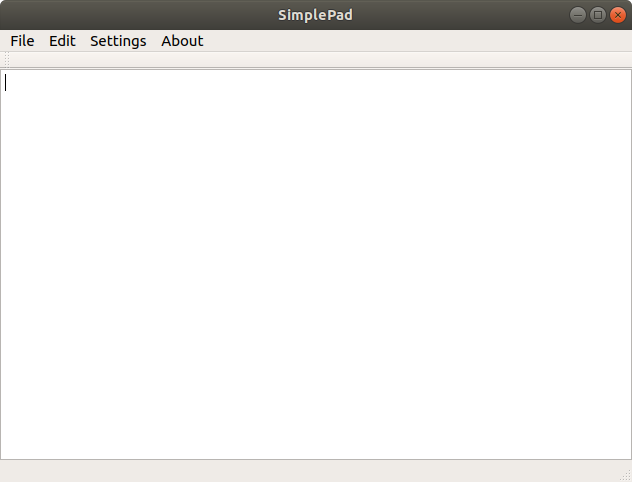
Šajā testa piemērā tiks demonstrētas visas teksta redaktora funkcijas.

## 4.1 Programmas darbības demonstrācija

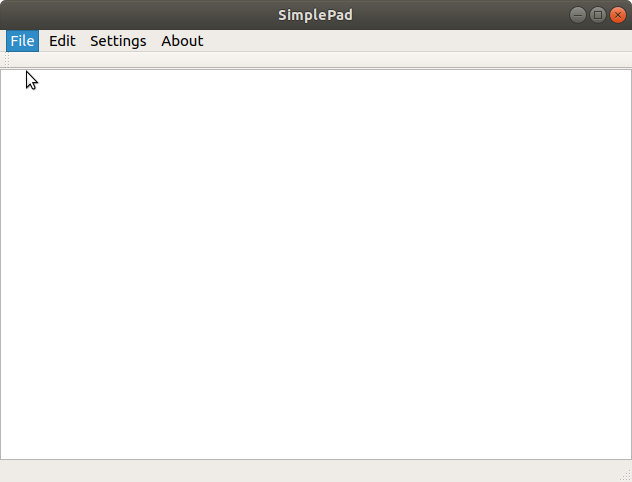
### 4.1.1 Programmas palaišaina

(2. attēls)

Šī c++ programma tiek palaista ievadot terminālī “$ ./kursa\_darbs\_cpp

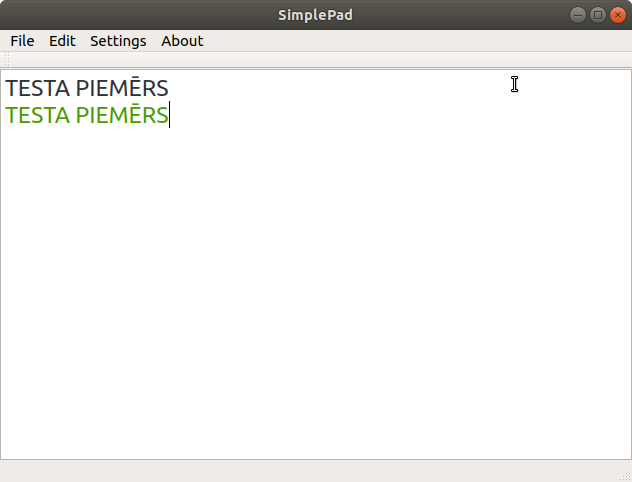
Kad programma atverās, parādās logs kā redzams 3. attēlā.(3. attēls)

### 4.1.2 Jauns fails

Jauns fails, jeb tīrs lauks, tiek izveidots izvēlētos rīkjoslas opciju “New file” (skatīt 4. attēlu.).(4. attēls)

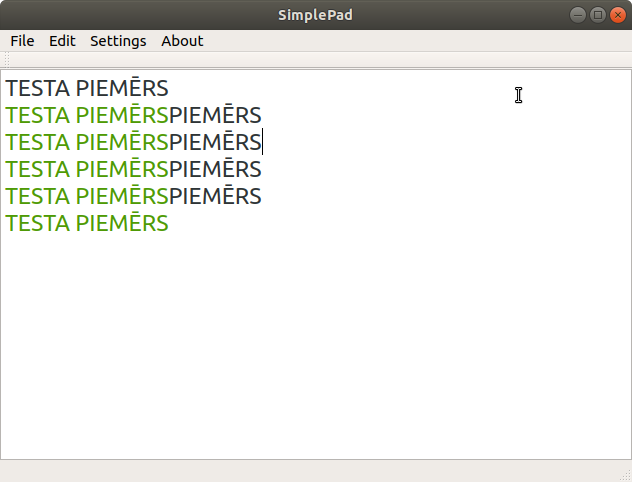
### 4.1.3 Teksta apstrāde

Šajā piemērā tiek ievadīts teksts, nomainīts tā fonts, uzspiežot uz rīkjoslas izvēlni “Settings” un pēc tam uz opciju “Change font”, nomainīta teksta krāsa, uzspiežot uz rīkjoslas izvēlni “Settings” un pēc tam izvēloties opciju “Change text color”(5. attēls)



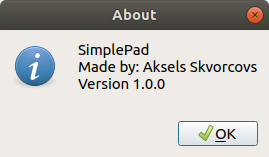
### 4.1.4 Teksta kopēšana un ielīmēšana

Šajā testa piemērā tiek iezīmēts teksts, un tad izmantotas rīkjoslas izvēlnes “Edit” piedāvātās opcijas. Sākumā tiek nospiests “Copy”, lai iezīmēto tekstu nokopētu. Pēc tam tiek izmantota opcija “Paste”, kas ielīmē nokopēto tekstu teksta redaktora rakstīšanas laukumā. Šajā piemērā tiek pārbaudītas arī pārējās “Edit” izvēlnes opcijas “Undo” un “Redo”. Tiek uzspiesta “Undo” opcija, kas atsauc pēdējo darbību, nodzēšot iepriekš ielīmēto tekstu. Pēc tam tiek uzspiesta “Redo”, kas izpilda atsaukto darbību, atgriežot nodzēsto ielīmēto tekstu atpakaļ. (Rezultātu skatīt 6. attēlā).

(6. attēls)

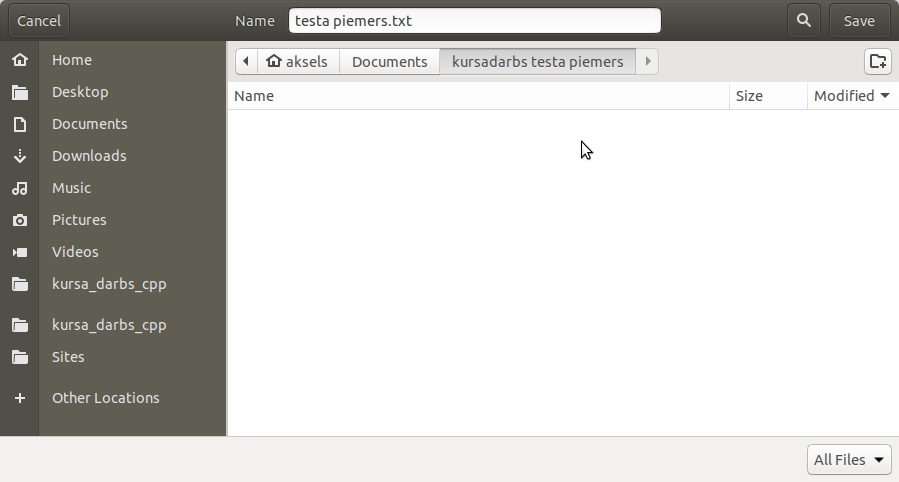
### 4.1.5 Informācijas logs

Uzspiežot uz “About” rīkjoslas izvēlni, atverās logs, kas izvada informāciju par lietojumprogrammu. To var aizvērt klikšķinot uz “OK” pogas. (Skatīt 7. attēlu).

(7. attēls)

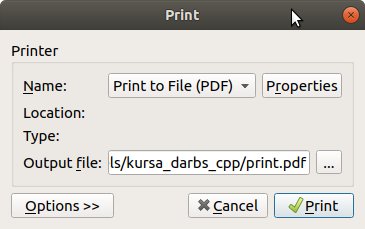
### 4.1.6 Saglabāt failu

Lai saglabātu failu, spiež uz “File” rīkjoslas izvēlnes, izvēloties “Save as” opciju. Atverās dialoga logs, kurš piedāvā jums saglabāt failu. Ievadot vēlamo nosaukumu un nospiežot “Save”, fails tiek saglabāts. (Skatīt 8. attēlu).

(8. attēls)

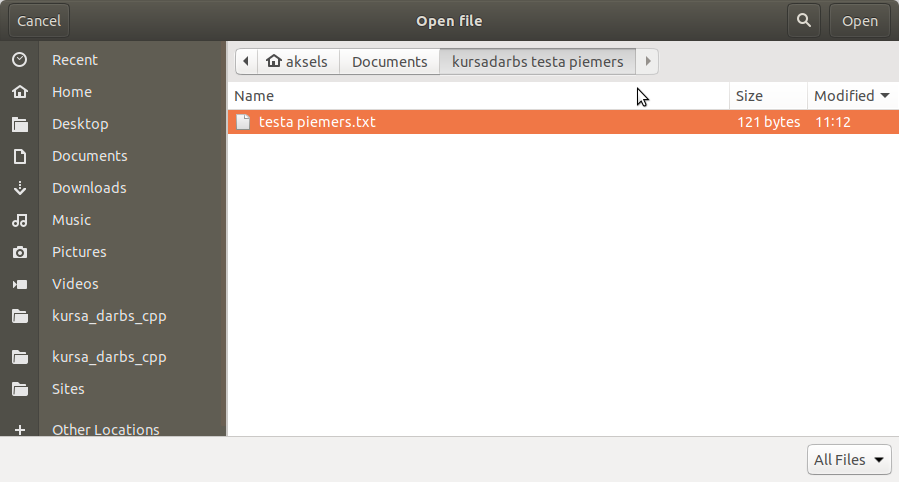
### 4.1.7 Printēt failu

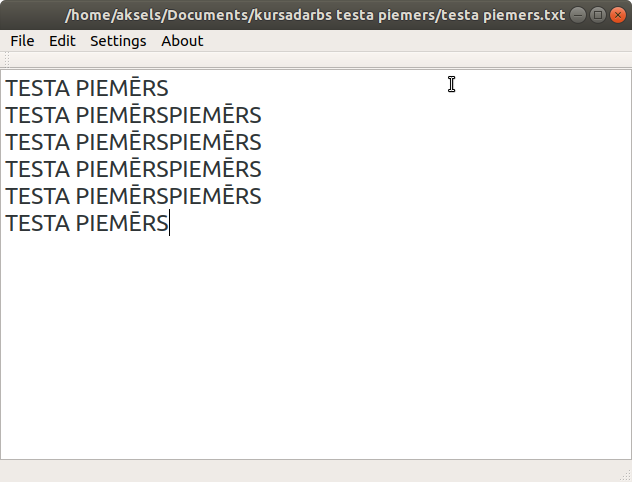
Ir izveidota un iespējota arī printēšanas opcija, bet to, diemžēl, nebija iespējams pārbaudīt, jo nav pieejams personīgais printeris. Uzspiežot uz rīkjoslas “File” un pēc tam “Print”, atverās printera izvēlnes dialoga logs. Sekojot tam, ir iespējams izprintēt faila tekstu uz lapas.

(9.attēls)

### 4.1.8 Faila atvēršana

Fail atvēršana sākas izvēloties rīkjoslas izvēlni “File”, un izvēloties opciju “Open”. Parādās failu izvēlnes dialoga logs, kurš piedāvā izvēlēties failu, kuru atvērt. Ir iespējams atvērt gandrīz jebkuru failu, bet tas nenozīmē, ka tas būs lasāms. Šajā piemērā mēs izvēlēsimies jau iepriekšējās demonstrācijās izveidoto failu “testa piemers.txt”. (Skatīt 10. attēlu.). Pievēršot uzmanību 11. attēlam, varam redzēt, ka fails ir bijis veiksmīgi saglabāts, un arī veiksmīgi atvērts.

(10. attēls)

(11. attēls)

# 5. Bibliogrāfiskais saraksts.

<https://wiki.qt.io/About_Qt>

<https://wiki.qt.io/Main>

<https://wiki.qt.io/Qt_for_Beginners>