

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ДИНАМИЧЕСКИЕ БИБЛИОТЕКИ (DLL) В QT CREATOR

**ОТЧЁТ ЛАБОРАТОРНОЙ РАБОТЫ №3**  
ПО ДИСЦИПЛИНЕ «ОПЕРАЦИОННЫЕ СИСТЕМЫ И СИСТЕМНОЕ  
ПРОГРАММИРОВАНИЕ»  
ЗА II СЕМЕСТР

Выполнил:  
студент 3-го курса  
V-го семестр  
факультета ЭИС  
группы ПО-4(1)  
зачётная книжка №190333  
Галанин П. И.  
«\_\_» \_\_\_\_\_ 2021 г.

Проверил:  
ассистент  
кафедры ИИТ  
Дряпко А. В.  
«\_\_» \_\_\_\_\_ 2021 г.

# Содержание

<b>1</b>	<b>Уточнение . . . . .</b>	<b>3</b>
<b>2</b>	<b>Задание (helper.dll) . . . . .</b>	<b>4</b>
2.1	Проект dll . . . . .	4
2.2	Проект gui . . . . .	7
<b>3</b>	<b>Задание (helper_class.dll) . . . . .</b>	<b>9</b>
3.1	Проект dll . . . . .	9
3.2	Проект gui . . . . .	10
<b>4</b>	<b>Задание (about.dll) . . . . .</b>	<b>11</b>
4.1	Проект dll . . . . .	11
4.2	Проект gui . . . . .	13
<b>5</b>	<b>Задание (изменение шрифта) . . . . .</b>	<b>14</b>
5.1	Проект gui . . . . .	14

Цель работы: научиться разрабатывать и использовать динамические библиотеки (DLL) с использованием Qt.

## 1 Уточнение

Чтобы подключить dll библиотеку, её нужно сбилдить в отдельном проекте, это можно делать как в Visual Studio 2022, создав шаблон Dynamic-Link Library (DLL), так и в Qt Creator, выбрав External Library.

Чтобы сбилдить проект в QtCreator выбираем справа сбоку проект, жмём правой кнопкой мыши и жмём «Build "...»>.

Проект создаст dll файл в папке «build-...debug» / «Название проекта» / «debug» / «dll файл», либо в «build-...release» / «Название проекта» / «release» / «dll файл».

Этот dll файл копируем рядом с \*.exe файлом графического приложения.

### Листинг: gpi\_copy\_dll.bat

@REM Копируем helper.dll

```
copy build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Debug\gpi_helper\
debug\gpi_helper.dll build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Debug\
gpi_osisp5_option5\debug\
```

```
copy build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Release\
gpi_helper\release\gpi_helper.dll build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Release\
gpi_osisp5_option5\release\
```

@REM Копируем about.dll

```
copy build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Debug\
gpi_about\debug\gpi_about.dll
build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Debug\
gpi_osisp5_option5\debug\
```

```
copy build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Release\
gpi_about\release\gpi_about.dll
build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Release\
gpi_osisp5_option5\release\
```

@REM Копируем helper\_class.dll

```
copy build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Debug\
gpi_helper_class\debug\gpi_helper_class.dll
build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Debug\
gpi_osisp5_option5\debug\
```

```
copy build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Release\
gpi_helper_class\release\gpi_helper_class.dll
build-gpi_osisp5_lab3-Desktop_Qt_6_2_2_MinGW_64_bit-Release\
gpi_osisp5_option5\release\
```

## 2 Задание (HELPER.DLL)

Выбрать 3 вспомогательные функции и вынести их описание и реализацию в динамическую библиотеку `helper.dll`. В основном приложении осуществить загрузку реализованных функций во время работы программы (at run-time, с использованием объекта `QLibrary`) и их вызов.

### 2.1 Проект dll

Листинг: `../gpi_osisp5_lab3/gpi_helper/gpi_helper.pro`

```
1  QT += widgets
2
3  TEMPLATE = lib
4  DEFINES += GPI_HELPER_LIBRARY
5
6  CONFIG += c++11
7
8  # You can make your code fail to compile if it uses deprecated APIs.
9  # In order to do so, uncomment the following line.
10 #DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs deprecated before Qt
    6.0.0
11
12 SOURCES += \
13     gpi_helper.cpp
14
15 HEADERS += \
16     gpi_helper_global.h \
17     gpi_helper.h
18
19 # Default rules for deployment.
20 unix {
21     target.path = /usr/lib
22 }
23 !isEmpty(target.path): INSTALLS += target
```

Листинг: `../gpi_osisp5_lab3/gpi_helper/gpi_helper_global.h`

```
1  #ifndef GPI_HELPER_GLOBAL_H
2  #define GPI_HELPER_GLOBAL_H
3
4  #include <QtCore/qglobal.h>
5
6  #if defined(GPI_HELPER_LIBRARY)
7  #   define GPI_HELPER_EXPORT Q_DECL_EXPORT
8  #else
9  #   define GPI_HELPER_EXPORT Q_DECL_IMPORT
10 #endif
11
12 #endif // GPI_HELPER_GLOBAL_H
```

## Листинг: ../gpi\_osisp5\_lab3/gpi\_helper/gpi\_helper.h

```
1 #ifndef GPI_HELPER_H
2 #define GPI_HELPER_H
3
4 #include "gpi_helper_global.h"
5 #include <QStringList>
6
7 extern "C" __declspec(dllexport)
8 QStringList gpi_get_1_level();
9
10 extern "C" __declspec(dllexport)
11 QStringList gpi_get_2_level();
12
13 extern "C" __declspec(dllexport)
14 QStringList gpi_get_3_level();
15
16 extern "C" __declspec(dllexport)
17 QStringList gpi_get_4_level();
18
19 #endif // GPI_HELPER_H
```

## Листинг: ../gpi\_osisp5\_lab3/gpi\_helper/gpi\_helper.cpp

```
1 #include "gpi_helper.h"
2
3 // gpi_ Функция, которая возвращает текстовый 1-ый уровень
4 QStringList gpi_get_1_level()
5 {
6     QStringList gpi_str_map;
7     // =====
8     gpi_str_map = {
9         "xxxxxxxxxxxxxxxxxx",
10        "xxxxxxxxxxxxxxxxxx",
11        "xxxxxxxxxxxxxxxxxx",
12        "xxxxxxxxxxxxxxxxxx",
13        "xxxxxx . . . xxxxxx",
14        "xxxxf@b . . xxxxxx",
15        "xxxxxx . bfxxxxxx",
16        "xxxxfxxb . xxxxxx",
17        "xxxx . x . f . xxxxxx",
18        "xxxxb . Bbbfxxxxx",
19        "xxxx . . . f . . xxxxx",
20        "xxxxxxxxxxxxxxxxxx",
21        "xBxxxxxxxxxxxxxxxx",
22        "xBxxxxxxxxxxxxxxxx",
23        "xBxxxxxxxxxxxxxxxx",
24        "xBxxxxxxxxxxxxxxxx",
25        "xxxxxxxxxxxxxxxxxx"
26    };
27     return gpi_str_map;
28 }
29
30 // gpi_ Функция, которая возвращает текстовый 2-ой уровень
31 QStringList gpi_get_2_level()
32 {
33     QStringList gpi_str_map;
34     // =====
35     gpi_str_map = {
36        "xxxxxxx . xxxxxxxxxxxxxxxxxxxxxxxxx",
37        "xxxxxxx . xxxxxxxxxxxxxxxxxxxxxxxxx",
38        "xx . . . xx . xx . . . . . . . . . . xx",
39        "xx . . . . x . x . . . . . . . . . . xx",
40        "xx . . . . . . . . . . bbbbbbbbbb . . . . . xx",
41    }
```

```

41         "xxx ..... xxx" ,
42         "xxxx ..... xxxx" ,
43         " .....@..... b..... " ,
44         "xxxx ..... fff ..... xxxx" ,
45         "xxx .... x ..... fff ..... xxx" ,
46         "xx ..... fff ..... xx" ,
47         "xx .... x.x ..... fff ..... xx" ,
48         "xx ... xx.xx ..... xx" ,
49         "xxxxxxxxxxxxxxxxxxxxxxxxxxxx" ,
50         "xxxxxxxxxxxxxxxxxxxxxxxxxxxx" ,
51     };
52     return gpi_str_map;
53 }
54
55 // gpi_ Функция, которая возвращает текстовый 3-ий уровень
56 QStringList gpi_get_3_level()
57 {
58     QStringList gpi_str_map;
59     // =====
60     gpi_str_map = {
61         "xxxxxxxxxxxxxxxx",
62         "xxxxxxxxxxxxxxxx",
63         "xxxxxxxxxxxxxxxx",
64         "xxxxxxxxxxxxxxxx",
65         "xxxxxx ... xxxxxx",
66         "xxxxf@b ... xxxxxx",
67         "xxxxxx . bfxxxxxx",
68         "xxxxfxxb . xxxxxx",
69         "xxxx . x . f . xxxxxx",
70         "xxxxb . Bbbfxxxxx",
71         "xxxx ... f . . xxxxxx",
72         "xxxxxxxxxxxxxxxx",
73         "xBxBxBxxxxxxxx",
74         "xBxBxBxxxxxxxx",
75         "xBxBxBxxxxxxxx",
76         "xBxBxBxxxxxxxx",
77         "xxxxxxxxxxxxxxxx"
78     };
79     return gpi_str_map;
80 }
81
82 // gpi_ Функция, которая возвращает текстовый 4-ый уровень
83 QStringList gpi_get_4_level()
84 {
85     QStringList gpi_str_map;
86     // =====
87     gpi_str_map = {
88         "xxxxxxxxxxxxxxxx",
89         "xxxxxxxxxxxxxxxx",
90         "xxxxxxxxxxxxxxxx",
91         "xxxxxxxxxxxxxxxx",
92         "xxxxxx ... xxxxxx",
93         "xxxxf@b ... xxxxxx",
94         "xxxxxx . bfxxxxxx",
95         "xxxxfxxb . xxxxxx",
96         "xxxx . x . f . xxxxxx",
97         "xxxxb . Bbbfxxxxx",
98         "xxxx ... f . . xxxxxx",
99         "xxxxxxxxxxxxxxxx",
100        "xBxBxBxxxxxxxx",
101        "xBxBxBxxxxxxxx",
102        "xBxBxBxxxxxxxx",
103        "xBxBxBxxxxxxxx",

```

```

104     "xxxxxxxxxxxxxxxxxx"
105 };
106 return gpi_str_map;
107 }

```

## 2.2 Проект gui

### Листинг: ../gpi\_osisp5\_lab3/gpi\_osisp5\_option5/gpi\_helper.cpp

```

1  #include "gpi_gamewindow.hpp"
2
3  // gpi_ Функция, которая рестратит уровень
4  void gpi_GameWindow::gpi_on_gpi_action_restart_triggered()
5  {
6      QLibrary gpi_lib("gpi_helper");
7
8      if (gpi_lib.load() == false)
9      {
10         QMessageBox::critical(this, "Error with dll", "Not found gpi_helper.dll");
11         qDebug() << "Not found dll gpi_helper.dll";
12         return;
13     }
14
15     typedef QStringList (*prototypeHelperDll)();
16
17     if (this->gpi_lvl == 1)
18     {
19         prototypeHelperDll gpi_get_1_level = (prototypeHelperDll)
gpi_lib.resolve("gpi_get_1_level");
20
21         if (gpi_get_1_level == NULL)
22         {
23             QMessageBox::critical(this, "Error with dll", "Not found function gpi_get_1_level()");
24             qDebug() << "Not found function gpi_get_1_level()";
25             return;
26         }
27
28         gpi_generate(gpi_get_1_level());
29         gpi_on_gpi_action_align_triggered();
30         return;
31     }
32
33     if (this->gpi_lvl == 2)
34     {
35         prototypeHelperDll gpi_get_2_level = (prototypeHelperDll)
gpi_lib.resolve("gpi_get_2_level");
36
37         if (gpi_get_2_level == NULL)
38         {
39             QMessageBox::critical(this, "Error with dll", "Not found function gpi_get_2_level()");
40             qDebug() << "Not found function gpi_get_2_level()";
41             return;
42         }
43
44         gpi_generate(gpi_get_2_level());
45         gpi_on_gpi_action_align_triggered();
46         return;
47     }
48
49     if (this->gpi_lvl == 3)
50     {
51         prototypeHelperDll gpi_get_3_level = (prototypeHelperDll)

```

```

52     gpi_lib.resolve("gpi_get_3_level");
53     if (gpi_get_3_level == NULL)
54     {
55         QMessageBox::critical(this, "Error with dll", "Not found function gpi_get_3_level()");
56         qDebug() << "Not found function gpi_get_3_level()";
57         return;
58     }
59
60     gpi_generate(gpi_get_3_level());
61     gpi_on_gpi_action_align_triggered();
62     return;
63 }
64
65 if (this->gpi_lvl == 4)
66 {
67     prototypeHelperDll gpi_get_4_level = (prototypeHelperDll)
gpi_lib.resolve("gpi_get_4_level");
68
69     if (gpi_get_4_level == NULL)
70     {
71         QMessageBox::critical(this, "Error with dll", "Not found function gpi_get_4_level()");
72         qDebug() << "Not found function gpi_get_4_level()";
73         return;
74     }
75
76     gpi_generate(gpi_get_4_level());
77     gpi_on_gpi_action_align_triggered();
78     return;
79 }
80
81 prototypeHelperDll gpi_get_1_level = (prototypeHelperDll) gpi_lib.resolve("gpi_get_1_level");
82
83 if (gpi_get_1_level == NULL)
84 {
85     QMessageBox::critical(this, "Error with dll", "Not found function gpi_get_1_level()");
86     qDebug() << "Not found function gpi_get_1_level()";
87     return;
88 }
89
90 gpi_generate(gpi_get_1_level());
91 gpi_on_gpi_action_align_triggered();
92 }

```



## 3 Задание (HELPER\_CLASS.DLL)

Выбрать вспомогательный класс и вынести его описание и реализацию в динамическую библиотеку `helper_class.dll`. В основном приложении осуществить загрузку реализованного класса во время компиляции (at compile-time).

### 3.1 Проект dll

Листинг: `../gpi_osisp5_lab3/gpi_helper_class/gpi_helper_class.pro`

```
1 QT -= gui
2
3 TEMPLATE = lib
4 DEFINES += GPI_HELPER_CLASS_LIBRARY
5
6 CONFIG += c++11
7
8 # You can make your code fail to compile if it uses deprecated APIs.
9 # In order to do so, uncomment the following line.
10 #DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs deprecated before Qt
    6.0.0
11
12 SOURCES += \
13     gpi_helper_class.cpp
14
15 HEADERS += \
16     gpi_helper_class_global.h \
17     gpi_helper_class.hpp
18
19 # Default rules for deployment.
20 unix {
21     target.path = /usr/lib
22 }
23 !isEmpty(target.path): INSTALLS += target
```

Листинг: `../gpi_osisp5_lab3/gpi_helper_class/gpi_helper_class_global.h`

```
1 #ifndef GPI_HELPER_CLASS_GLOBAL_H
2 #define GPI_HELPER_CLASS_GLOBAL_H
3
4 #include <QtCore/qglobal.h>
5
6 #if defined(GPI_HELPER_CLASS_LIBRARY)
7 #   define GPI_HELPER_CLASS_EXPORT Q_DECL_EXPORT
8 #else
9 #   define GPI_HELPER_CLASS_EXPORT Q_DECL_IMPORT
10 #endif
11
12 #endif // GPI_HELPER_CLASS_GLOBAL_H
```

Листинг: `../gpi_osisp5_lab3/gpi_helper_class/gpi_helper_class.hpp`

```
1 #ifndef GPI_HELPER_CLASS_HPP
```

```

2 #define GPI_HELPER_CLASS_HPP
3
4 #include "gpi_helper_class_global.h"
5 #include <QDebug>
6
7 class GPI_HELPER_CLASS_EXPORT gpi_helper_class
8 {
9 public:
10     gpi_helper_class();
11 };
12
13 extern "C" GPI_HELPER_CLASS_EXPORT gpi_helper_class* gpi_create_gpi_helper_class()
14 {
15     return new gpi_helper_class();
16 }
17
18 #endif // GPI_HELPER_CLASS_HPP

```

**Листинг: ../gpi\_osisp5\_lab3/gpi\_helper\_class/gpi\_helper\_class.cpp**

```

1 #include "gpi_helper_class.hpp"
2
3 gpi_helper_class::gpi_helper_class()
4 {
5     qDebug() << " gpi_helper_class.dll constructor";
6 }

```

## 3.2 Проект gui

**Листинг: ../gpi\_osisp5\_lab3/gpi\_osisp5\_option5/gpi\_helper\_class.hpp**

```

1 #ifndef GPI_HELPER_CLASS_HPP
2 #define GPI_HELPER_CLASS_HPP
3
4 #include <QLibrary>
5 #include <QMessageBox>
6 #include <QDebug>
7
8 class gpi_helper_class
9 {
10 public:
11     gpi_helper_class();
12 };
13
14 void gpi_load_dll_gpi_helper_class();
15
16 #endif // GPI_HELPER_CLASS_HPP

```

**Листинг: ../gpi\_osisp5\_lab3/gpi\_osisp5\_option5/gpi\_helper\_class.cpp**

```

1 #include "gpi_helper_class.hpp"
2
3 void gpi_load_dll_gpi_helper_class()
4 {
5     QLibrary gpi_lib("gpi_helper_class");
6
7     if (gpi_lib.load() == false)
8     {
9         QMessageBox gpi_mb;
10         gpi_mb.setWindowTitle("Error with dll");
11         gpi_mb.setText("Not found gpi_helper_class.dll");

```

```

12     gpi_mb.exec();
13     qDebug() << "Not found dll gpi_helper_class.dll";
14     return;
15 }
16
17 typedef gpi_helper_class* (*prototypeHelperClass)();
18 prototypeHelperClass gpi_helper_class = (prototypeHelperClass)
gpi_lib.resolve("gpi_create_gpi_helper_class");
19
20 if (gpi_helper_class == NULL)
21 {
22     QMessageBox gpi_mb;
23     gpi_mb.setWindowTitle("Error with dll");
24     gpi_mb.setText("Not found function gpi_helper_class()");
25     gpi_mb.exec();
26     qDebug() << "Not found function gpi_helper_class()";
27     return;
28 }
29
30 qDebug() << " Loaded gpi_helper_class()";
31 }

```

## 4 Задание (ABOUT.DLL)

Реализовать окно «О программе» в виде объекта динамической библиотеки about.dll с указанием автора программы, группы, курса и краткого описания разработанного приложения.

Осуществить импорт указанной библиотеки и отображение соответствующего окна при выборе пункта меню «О программе».

### 4.1 Проект dll

Листинг: ../gpi\_osisp5\_lab3/gpi\_about/gpi\_about.pro

```

1 QT += widgets
2
3 TEMPLATE = lib
4 DEFINES += GPI_ABOUT_LIBRARY
5
6 CONFIG += c++11
7
8 # You can make your code fail to compile if it uses deprecated APIs.
9 # In order to do so, uncomment the following line.
10 #DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs deprecated before Qt
    6.0.0
11
12 SOURCES += \
13     gpi_about.cpp
14
15 HEADERS += \
16     gpi_about_global.h \
17     gpi_about.hpp
18
19 # Default rules for deployment.

```

```

20  unix {
21      target.path = /usr/lib
22  }
23  !isEmpty(target.path): INSTALLS += target

```

### Листинг: ../gpi\_osisp5\_lab3/gpi\_about/gpi\_about\_global.h

```

1  #ifndef GPI_ABOUT_GLOBAL_H
2  #define GPI_ABOUT_GLOBAL_H
3
4  #include <QtCore/qglobal.h>
5
6  #if defined(GPI_ABOUT_LIBRARY)
7  #   define GPI_ABOUT_EXPORT Q_DECL_EXPORT
8  #else
9  #   define GPI_ABOUT_EXPORT Q_DECL_IMPORT
10 #endif
11
12 #endif // GPI_ABOUT_GLOBAL_H

```

### Листинг: ../gpi\_osisp5\_lab3/gpi\_about/gpi\_about.hpp

```

1  #ifndef GPI_ABOUT_HPP
2  #define GPI_ABOUT_HPP
3
4  #include "gpi_about_global.h"
5  #include <QMessageBox>
6
7  extern "C" __declspec(dllexport)
8  void gpi_about();
9
10 #endif // GPI_ABOUT_HPP

```

### Листинг: ../gpi\_osisp5\_lab3/gpi\_about/gpi\_about.cpp

```

1  #include "gpi_about.hpp"
2
3  void gpi_about()
4  {
5      QMessageBox gpi_mb;
6      gpi_mb.setWindowTitle("О программе");
7      gpi_mb.setText(
8          "Программу разработал \n"
9          "студент 3-его курса \n"
10         "факультета ЭИС \n"
11         "группы ПО-4(1) \n"
12         "Галанин Павел \n"
13         "(декабрь 2021) \n"
14         " \n"
15         "Хорошей игры! \n"
16     );
17     gpi_mb.exec();
18 }

```

## 4.2 Проект gui

Листинг: ../gpi\_osisp5\_lab3/gpi\_osisp5\_option5/gpi\_about.cpp

```
1  #include "gpi_mainwindow.hpp"
2  #include "gpi_gamewindow.hpp"
3
4  void gpi_MainWindow::gpi_on_gpi_action_about_triggered()
5  {
6      QLibrary gpi_lib("gpi_about");
7
8      if (gpi_lib.load() == false)
9      {
10         QMessageBox::critical(this, "Error with dll", "Not found gpi_helper.dll");
11         qDebug() << "Not found dll gpi_helper.dll";
12         return;
13     }
14
15     typedef QStringList (*prototypeHelperDll)();
16
17     prototypeHelperDll gpi_about = (prototypeHelperDll) gpi_lib.resolve("gpi_about");
18
19     if (gpi_about == NULL)
20     {
21         QMessageBox::critical(this, "Error with dll", "Not found function gpi_about()");
22         qDebug() << "Not found function gpi_about()";
23         return;
24     }
25
26     gpi_about();
27 }
28
29 void gpi_GameWindow::gpi_on_gpi_action_about_triggered()
30 {
31     QLibrary gpi_lib("gpi_about");
32
33     if (gpi_lib.load() == false)
34     {
35         QMessageBox::critical(this, "Error with dll", "Not found gpi_helper.dll");
36         qDebug() << "Not found dll gpi_helper.dll";
37         return;
38     }
39
40     typedef QStringList (*prototypeHelperDll)();
41
42     prototypeHelperDll gpi_about = (prototypeHelperDll) gpi_lib.resolve("gpi_about");
43
44     if (gpi_about == NULL)
45     {
46         QMessageBox::critical(this, "Error with dll", "Not found function gpi_about()");
47         qDebug() << "Not found function gpi_about()";
48         return;
49     }
50
51     gpi_about();
52 }
```

## 5 Задание (изменение шрифта)

Реализовать расширения для приложения, позволяющие изменять оформление пунктов меню (шрифт, размер, начертание и т.д.). Соответствующие изменения должны происходить при выборе специального пункта меню. Создать как минимум три расширения такого типа.

### 5.1 Проект gui

Листинг: ../gpi\_osisp5\_lab3/gpi\_osisp5\_option5/gpi\_font.cpp

```
1  #include "gpi_mainwindow.hpp"
2  #include "ui_gpi_mainwindow.h"
3
4  void gpi_MainWindow::gpi_on_gpi_action_font_triggered()
5  {
6      bool gpi_ok;
7      QFont gpi_newFont;
8      // =====
9
10     gpi_newFont = QFontDialog::getFont(&gpi_ok, this);
11
12     if (gpi_ok)
13     {
14         ui->centralwidget->setFont(gpi_newFont);
15         QSettings settings(this);
16         settings.setValue("gpi_VIEWWF", gpi_newFont);
17     }
18 }
```

**Вывод:** научились использовать dll в QtCreator.