

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

09.03.04 – Программная инженерия

Профиль направления подготовки бакалавриата

Системное и прикладное программное обеспечение

Отчет о проектной работе по курсу "Основы информатики и программирования

РАЗРАБОТКА ПРИЛОЖЕНИЯ "МУЗЫКАЛЬНЫЙ ПЛЕЕР

Выполнил:

студент 1 курса группы 22107

Н.Н. Анисимов

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Требования</b>	<b>3</b>
<b>2 Проектирование</b>	<b>3</b>
2.1 Общая структура . . . . .	3
2.2 Выбор папки и поиск mp3 файлов . . . . .	3
2.3 Хранение и обновление списка . . . . .	3
2.4 Воспроизведение и пауза . . . . .	3
<b>3 Реализация</b>	<b>4</b>
<b>4 Заключение</b>	<b>7</b>

# Введение

Моей задачей было разработать простой музыкальный плеер, в котором можно было бы открывать папки с музыкой и проигрывать ее.

## 1 Требования

Приложение должно давать выбрать папку и затем выводить список содержащихся там mp3 файлов. Файлы можно будет воспроизводить, а также ставить на паузу. В строке состояния будет выводиться текущая папка.

## 2 Проектирование

### 2.1 Общая структура

За список mp3 файлов, его хранение и обновление, а также за воспроизведение и паузу аудиофайла будет отвечать объект `SongList`.

За вывод на экран, строку состояний, кнопки будет отвечать объект `MainWindow`.

### 2.2 Выбор папки и поиск mp3 файлов

Для выбора папки будем использовать `QFileDialog`, для поиска mp3 файлов - `QDirIterator` вместе с `QRegExp`.

### 2.3 Хранение и обновление списка

Будем хранить песни в `QStringList`. Так же папку, которую необходимо проверить в `SongList`. Доступ к песням можно получить с помощью функций `hasNext()` и `next()`. Установить папку - `set_dir()`. Обновить список - `refresh()`.

### 2.4 Воспроизведение и пауза

В `SongList` будем хранить указатель на `QMediaPlayer`, который будем создавать при включении какой-нибудь песни. Функция `play()` будет отвечать за включение новой песни, а `pause()` за паузу и снятие с паузы. Будет так же хранить указатель на текущую песню - `active_song`.

Сделаем слот `change()` чтобы менять иконку рядом с песней в списке при смене ее состояния.

### 3 Реализация

#### Листинг 1 – Кнопки

---

```
void MainWindow::on_open_clicked()
{
    QFileDialog dialog(this);
    dialog.setFileMode(QFileDialog::Directory);
    QStringList fileNames;
    if (dialog.exec())
        fileNames = dialog.selectedFiles();
    if(fileNames.empty()) return;
    statusBar()->showMessage(fileNames.at(0));
    songlist.set_dir(fileNames);
    on_refresh_clicked();
}

void MainWindow::on_refresh_clicked()
{
    ui->snglst->clear();
    songlist.refresh();
    QString sng;
    while(songlist.hasNext())
    {
        sng = songlist.next();
        QListWidgetItem* tmp = new QListWidgetItem(QIcon(":/play.png"),sng);
        ui->snglst->addItem(tmp);
    }
}
```

---

```
void SongList::refresh()
{
    songs.clear();
    QDirIterator go(dir,QDirIterator::NoIteratorFlags);
    while(go.hasNext())
    {
        QFile f(go.next());
        QFileInfo fi(f);
        if(!fi.isFile()) continue;
        QRegExp form("*.mp3");
        form.setPatternSyntax(QRegExp::Wildcard);
        if(form.exactMatch(f.fileName()))
        {
            songs.append(fi.baseName());
        }
    }
    songs.sort();
    songiter = -1;
    if(player!=NULL) delete player;
    player = NULL;
    active_song = NULL;
}
```

---

```
void SongList::play(QListWidgetItem* sng)
{
    if(player!=NULL)
    {
        if(is_played) player->pause();
        delete player;
    }
    player = new QMediaPlayer;
    player->setMedia(QUrl::fromLocalFile(
    QDir::toNativeSeparators(dir+"/"+sng->text()+".mp3")));
    player->setVolume(100);
    is_played = 0;
    active_song = sng;
    QObject::connect(player,&QMediaPlayer::stateChanged,this,&SongList::change);
    player->play();
}
void SongList::pause(QListWidgetItem* sng)
{
    if(active_song!=sng) return;
    if(is_played)
        player->pause();
    else
        player->play();
}
void SongList::change()
{
    if(is_played)
        active_song->setIcon(QIcon(":/play.png"));
    else
        active_song->setIcon(QIcon(":/pause.png"));
    is_played = (is_played)?0:1;
}
```

---

## 4 Заключение

У меня получилось разработать необходимое приложение, отвечающее всем требованиям.