

其中, selectedFiles用来返回用户选择的文件, 默认情况下只允许选择该文件。我们把文件路径存放在QStringList变量中, 然后第一个元素就是所选文件的路径名。

为“选择多个的文件打开对话框”按钮添加单击按钮事件处理函数, 代码如下:

```
void Dialog::on_pushButton_3_clicked()
{
    QFileDialog dlg(this, NULL, NULL, z("文本文件 (*.txt);;所有文件 (*.*)"));
    dlg.setDirectory("d:\\"); //设置初始路径
    dlg.setFileMode(QFileDialog::ExistingFiles); //设置允许选择多个已存在的文件
    if(dlg.exec() == QFileDialog::Accepted)
    {
        QStringList str_path_list = dlg.selectedFiles(); //返回所选文件路径名
        for (int i = 0; i < str_path_list.size(); i++)
        {
            QString str_path = str_path_list[i]; //依次得到单个文件路径
            // 显示文件路径
            QMessageBox::information(this, "caption", str_path);
        }
    }
}
```

这里故意调用函数setDirectory()来设置目录对话框显示时所定位的初始路径。

为“文件保存对话框”按钮添加单击按钮事件处理函数, 代码如下:

```
void Dialog::on_pushButton_2_clicked()
{
    QFileDialog dlg(this, NULL, "d:\\", z("文本文件 (*.txt);;所有文件 (*.*)"));
    dlg.setAcceptMode(QFileDialog::AcceptSave); //设置文件对话框为文件保存对话框
    if(dlg.exec() == QFileDialog::Accepted)
    {
        QStringList l = dlg.selectedFiles();
        QString str = l[0];
        QMessageBox::information(this, "caption", str); //显示文件路径
    }
}
```

(4) 保存项目并运行, 运行结果如图4-48所示。

4.10.3 字体对话框QFontDialog类

字体对话框可以让用户选择字体的字符集、字体大小、是否斜体/粗体等属性。Qt提供了QFontDialog类来实现字体对话框。

利用QFontDialog类显示字体对话框时首先要包含头文件:

```
#include <QFontDialog>
```

显示字体对话框有两种方法: 一种是使用静态函数; 另外一种定义对话框对象, 然后调用exec()函数。

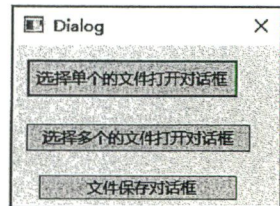


图4-48