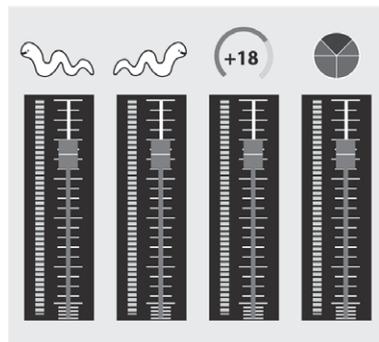


操作しやすい GUIアプリ作り

第3回 デスクトップにメッセージを
フレーム・レスで置いておく

澤田 英宏



(a) ベース



(b) フレーム・レス



(c) メッセージのみを表示 (部品部分を非表示) にした場合
図1 制作するのはフレーム・レスのメッセージ

作るもの

前回はウィンドウ内に部品を配置するレイアウト・クラスを紹介しましたが、今回は、部品を配置していくつかの機能を持ったアプリを作成します。

通常のGUIアプリでは、図1(a)のようなウィンドウ枠を持ったアプリになりますが、今回は任意のメッセージを表示する目的のため、起動時には図1(b)のようなフレーム・レスとなり、その後、「非表示」のラジオ・ボタンを押すと、メッセージだけを表示するウィンドウに切り替わります[図1(c)]。

リスト1 メイン・プログラム main.py

```
000: import sys
001: from PySide6.QtWidgets import QApplication
002: from MainWindow import MainWindow
003:
004: def __main__():
005:     app = QApplication()
006:     w = MainWindow()
007:     w.show()
008:     sys.exit(app.exec())
009:
010: if __name__ == '__main__':
011:     __main__()
```

アプリとしては単純な使用方法なのでプログラムを確認する前に使ってみてください。

開発環境の準備

今回も統合開発環境にはJetBrains社のPyCharmを使っています。開発環境の構築については、本誌サポート・ページで案内します。

<https://interface.cqpub.co.jp/inst-pycharm/>

ファイルは、プロジェクトを新規作成して、main.py(リスト1)とMainWindow.py(リスト2)の2つのファイルを作成します。これらのファイルは本書のサポート・ページから入手できます。

作り方

● 全体のレイアウト

部品配置にはレイアウト・クラスのうちの1つである「QGridLayout」を使用しました。QGridLayoutはウィジェットをグリッド状にレイアウトします。各部品の配置イメージは図2です。「label」「Line Edit」「Group Box」はマスをまたいで配置しました。

第2回で「QGridLayout」の配置には座標を用いることを紹介しましたが、ここでも再度確認しておきます。実際のコードの部分は後で確認するとして、ここでは各部品の配置座標だけを確認することにします。