14章 アプリケーションを作る

12T4069L
佐鳥 恭太郎

アプリケーションを作成する手順

- アプリケーションに必要な部品（ウィンドウ,ボタン,etc…）を考える
- データ構造の実現
- データを管理するシステムの実現
- 必用ならばGUIの開発

データ構造の特殊メソッド

・スーパークラスにobjectを継承したときに使用できる特殊メソッド（一部）の使い方

特殊メソッド名	説明
__repr__(self)	組み込み関数repr()や、文字列への変換時に呼び出される。戻り値は文字列。
__len__(self)	組み込み関数len()を実現するために呼び出される。戻り値はオブジェクトの長さ0以上。
__getitem__(self,key)	self[key]の値評価を実現するために呼び出される。スライスにも対応する。
__setitem__(self,key,value)	self[key]に対する代入を実現するために呼び出される。
__delitem__(self,key)	self[key]の削除を実現するために呼び出される。
__add__(self[,...])	二項算術演算+をエミュレートする。
__iadd__(self[,...])	累算算術代入+=をエミュレートする。

GUIの開発

- 標準ライブラリのTkinter（tk interface）というモジュールを使用する。
  - 部品構成がFrameとWidgetに分かれている。
  - Frame:だいたいウィンドウのこと
  - Widget:ボタン,チェックボックス,テキストなど…

ウィンドウを作る

```
>>> import Tkinter # Tkinterモジュールのインポート
>>> root = Tkinter.Tk() # トップレベルウィジェット作成（ウィンドウ）
>>> root.mainloop() # ウィンドウのメインループの開始
```

- mainloop()を呼ぶことでユーザの操作を受け付けられるようになる。
- ただし、標準入力が出来なくなる。
- tkinterのウィンドウを終了すると標準入力ができるようになる。

ボタンを作成してみる

ボタンオブジェクトのインスタンスを作成する

```
>>> root = Tkinter.Tk()
>>> b = Tkinter.Button(root[,設定,...]) # ボタン作成
>>> b.pack() # ボタンを設置する
```

- rootにボタンbを作成する。
- 設定が必要な時は第2引数以降に記述する。

ウィジェットの設定を変更する

- 作成済みのウィジェットの設定を変更する

>>> ウィジェットインスタンス.config(設定=値[,...])

設定と値は辞書オブジェクトでも渡せる

>>>ウィジェットインスタンス.config({“設定”:値[,...]})

ボタンのプログラム

```
# quitと書かれたボタンを作成する。押すとウィンドウが閉じる。
>>> from Tkinter import *
>>> root = Tk()
>>> b = Button(root, text="quit", command=root.destroy)
>>> b.pack()
>>> root.mainloop()
```

- シェルで記述して動作させるとウィンドウが生成され、ウィジェットが配置される様子が見れる。
- プログラムとして実行するとウィンドウはmainloop()で表示される。

Widgetの例

- tkinterで利用できるWidgetの例

Widgetのクラス名	説明
Label	テキストのラベル
Button	ボタン
Entry	1行テキスト入力フィールド
Checkbutton	チェックボタン
Listbox	リストボックス
Menu	メニュー
Radiobutton	ラジオボタン
Scale	スケール（スライダ）
Scrollbar	スクロールバー
Text	複数行テキスト入力
Canvas	画像などを表示する部品

イベント処理

- ユーザがWidgetやFrameに対して操作を行った時、イベントが発生する。

- 特定のイベントが発生した時、関数を実行したいときは、以下のbind()関数で設定する。

>>> オブジェクト.bind(“<イベント名>”, イベント関数名)

- イベントが発生した時、設定した関数にイベントオブジェクトが渡されるので、イベントの詳細はそのオブジェクトを読み取る。

GUIプログラムの流れ

```
>>> from Tkinter import *
>>> def イベント関数(event) : …
>>> root = Tk([設定,…]) # トップレベルウィジェット(親ウィンドウ)
>>> インスタンス = ウィジェット(root[,設定,…])
>>> インスタンス.bind(“<イベント名>”, イベント関数名)
>>> インスタンス.pack([設定,…]) # ウィジェット配置
>>> …
>>> root.mainloop() # ループ開始
>>> # プログラム終了処理
```

課題

- キーボードのキーが押されたときに何が押されたのかをフレームにテキスト（ラベル）で表示する。