

レポート提出の練習 (第 1 回)

氏名	入佐 啓士
クラス	理科 1 類 37 組
学生証番号	J4-220897

課題1.0 - 1.3節 例 1: x線の描画 crossLine.py

プログラムリスト

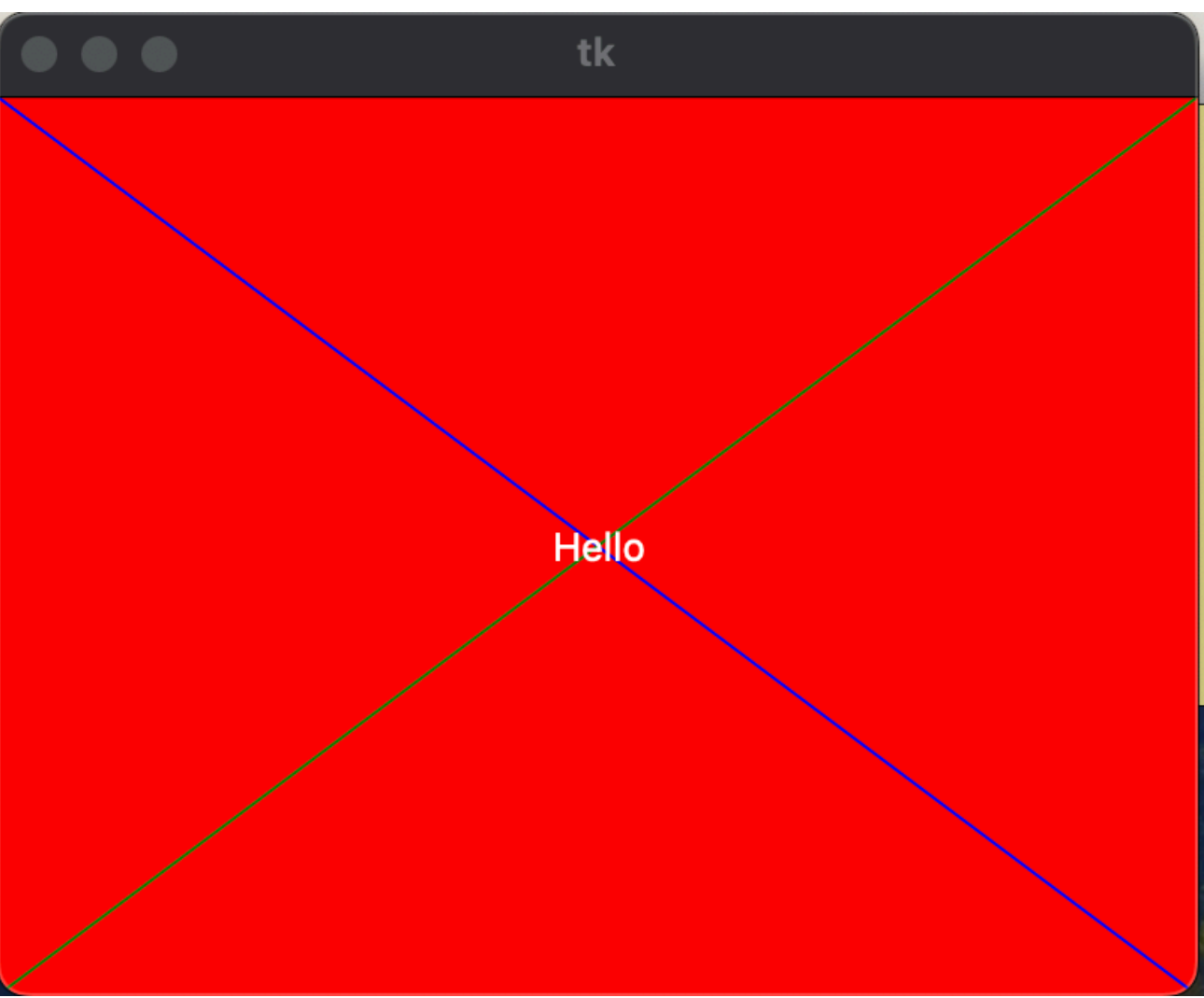
```
1      from tkinter import *                                # tkinterモジュールのimport
2      #from tkinter import Tk, Canvas                      # tkinterモジュールからTk,Canvasのimport
3      import sys                                           # sysモジュールのimport
4
5      W, H = (400, 300)                                    # canvasの幅と高さ
6
7      def display(canvas, msg):                             # 描画関数
8          '''
9          canvas - 描画するcanvas
10         2本の線分と文字列を描画する
11         '''
12         canvas.create_line((0, 0), (W-1, H-1), fill = 'blue') # 線分の描画 (左上→右下)
13         canvas.create_line((0, H-1), (W-1, 0), fill = 'green') # 線分の描画 (左下→右上)
14         canvas.create_text((W/2, H/2), text=msg) # 文字列の描画 (canvas中央)
15
16     def main():                                            # main関数
17         if len(sys.argv) > 1:                             # シェル引数がある場合
18             msg = sys.argv[1]                             # 第1引数を頂点数の文字列
19         else:                                             # シェル引数がない場合
20             msg = input('message -> ')                   # 描画する文字列を入力
21         root = Tk()                                       # ルートフレームの作成
22         canvas = Canvas(root, width = W, height = H, bg = 'red', highlightthickness=0) # canvasの作成
23
24         canvas.pack()                                    # canvasの配置確定
25         display(canvas, msg)                             # 描画関数 (display) の呼出
26         root.mainloop()                                 # ルートフレームの実行ループ開始
27
28     if __name__ == '__main__':                           # 起動の確認 (コマンドラインからの起動)
29         main()                                           # main関数の呼出
```

実行コマンド

```
$ python crossLine.py "Hello"
```

実行結果

```
(文字列の表示なし)
```



考察

描画領域において対角線2本を記述するプログラムを走らせた。例題と同じプログラムコードを実行したのにもかかわらず、先生の実行結果と描画領域の色が違っている（先生は白、私は黒）ことが気になったため、領域の色を変えることに取り組もうと思った。canvasオブジェクトの色のパラメタを指定しなかった場合、コンピュータのデフォルトの色設定が適用されると推測したため、22行目にbgパラメタ'red'を追加することにした。その結果、描画領域の色は赤に変化した。また描画領域の色をパラメタによって変化させることができたため、他のオブジェクトもパラメタによって色が変化させられると推測し、対角線の色を変化させようと考えた。初めはcanvasオブジェクトと同じようにbg='blue'で指定したが、うまく動作しなかった。調べてみるとオブジェクトごとに色指定のパラメタが異なり、対角線の場合はfillで指定することがわかり、12,13行目にfill = 'blue'とfill = 'green'を追加した結果、対角線の色はそれぞれ青、緑に変化した。

課題1.0 - 1.3節 例 2: X線の描画 crossLines.py

プログラムリスト

```
1      from crossLine import main
2      import sys
3
4      # 引数にint型の回数が指定されていることを確認
5      if len(sys.argv) <= 1:
6          count = int(input("count - >"))
7      elif sys.argv[2].isdigit():
8          count = int(sys.argv[2])
9      else:
10         count = int(input("count - >"))
11
12
13     if __name__ == '__main__':
14         for i in range(count):
15             main()
16             print("Done!")
```

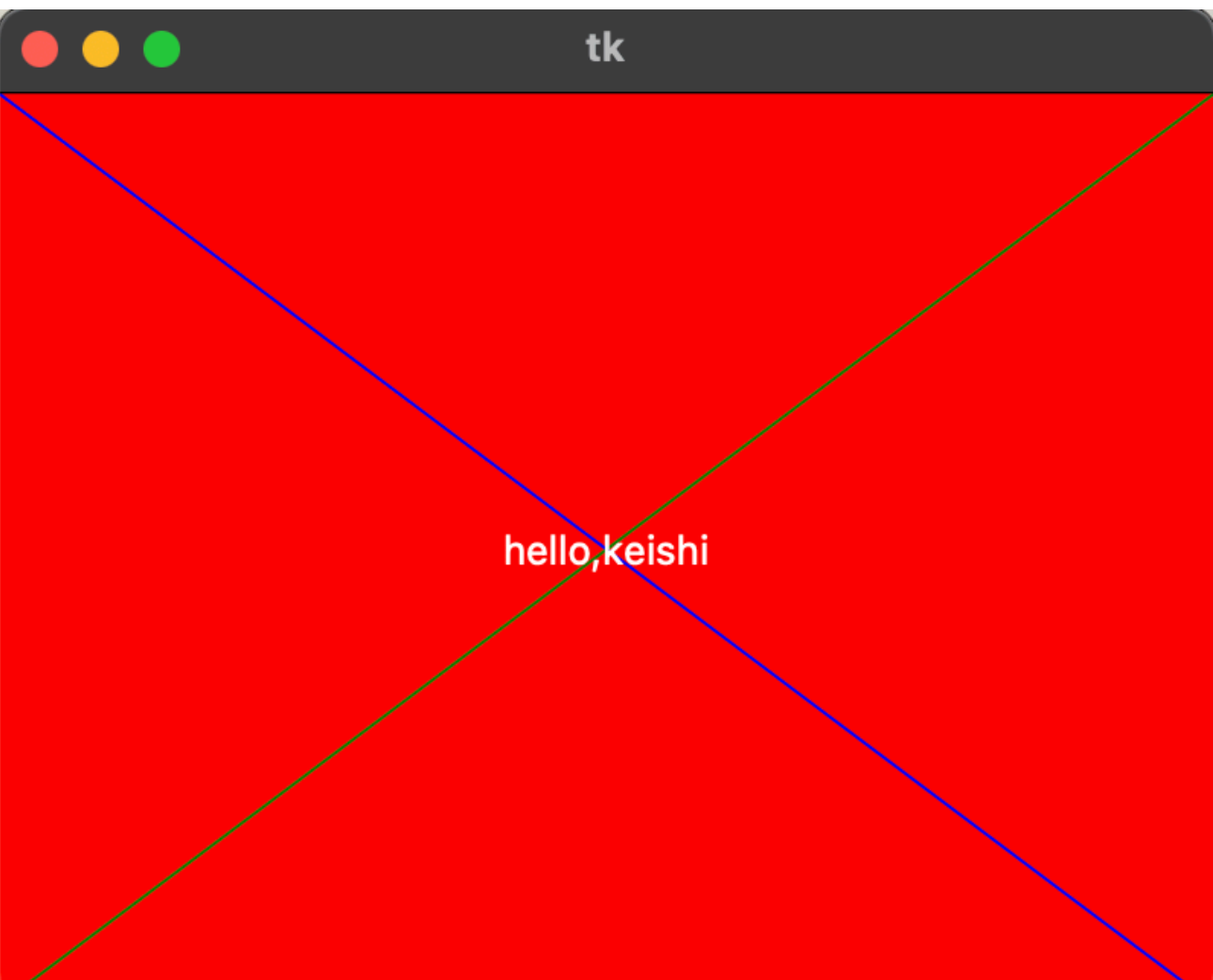
実行コマンド

```
$ python crossLines.py "hello,keishi" 3
```

実行結果

```
(base) irisakeishi@Irisas-MacBook-Air Chap01 % python crossLines.py "hello,keishi" 3
Done!
Done!
Done!
```

ウィンドウが表示された後、ウィンドウを削除するとDone!が表示され、その工程が3回繰り返された。



考察

例題ではmain()を3回記述されていたが、for文で記述することによりmain()の処理回数を変化させられると考えた。処理回数を指定するためにinputを用いようと最初は考えたが、例題1でargvを使用しているのを思い出し、同じように回数も引数で指定できるのではないかと考えた。まず、2行目でsysをimportしてargvによる処理をできる状態にした。次に5~10行目で第2引数に回数を指定しているかの処理を記述して、実行した結果intではない型はrange()に入れられませんとエラーメッセージが出た。そこでint()の処理により正しくint型に置換し、またint型ではなかった場合にcountの指定を求めるように変更した。その結果正しくcountで指定した回数main()を繰り返せるようになった。また正しく指定した回数繰り返せているかを確認するためにprint関数を使用した。

課題や授業に関して

レポート作成に要した時間

およそ1時間

特に苦労した点

例題2の5~10行目の第二引数による処理を記述するのに時間がかかった。ベン図を書いてelse,elifの中で必要な処理が考えやすくなるように工夫した。

授業についての感想や希望

zipファイルへの処理の仕方や、htmlファイルをsafariで見する方法など、コーディング以外の知識などを丁寧に教えていただけるのがありがたいです。初学者にとって環境構築やターミナルの扱い方などコーディング以外で詰まってしまうことがあると思うので、引き続きコーディング以外の操作も丁寧に教えていただきたいです。