**【課題 17：stock_prices.py】**

配布した ufj.html から，正規表現を用いて日付と株価を抽出し，これらを属性に持つ StockPrice オブジェクトを要素とするリストを生成するコードを実装せよ．ufj.html は図 1 のソースコードである．

東証PRM　NYSE (5.13ドル)

8306 銀行業　　リアルタイム株価 15:00

(株)三菱ＵＦＪフィナンシャル・グループ　719.5

ポートフォリオに追加　　前日比 -3 (-0.42%)

詳細情報 | チャート | 時系列 | ニュース | 企業情報 | 掲示板 | 株主優待 | 業績予報 | みんかぶ

株価時系列　信用残時系列

年初来高値 828 (2022年3月23日) | 年初来安値 630 (2022年1月4日)

2020/11/9 から 2022/11/11 を 日間 表示　初期設定に戻す

| 日付 | 始値 | 高値 | 安値 | 終値 | 出来高 | 調整後終値* |
|---|---|---|---|---|---|---|
| 2022年11月11日 | 715 | 716.4 | 706.1 | 710.8 | 47,598,200 | 710.8 |
| 2022年11月10日 | 707.3 | 711.3 | 706.4 | 710.7 | 28,718,200 | 710.7 |
| 2022年11月9日 | 710 | 712.6 | 706.4 | 708.5 | 33,052,600 | 708.5 |
| 2022年11月8日 | 707.3 | 714 | 704.9 | 710.8 | 42,531,800 | 710.8 |
| 2022年11月7日 | 708 | 709 | 703.3 | 703.8 | 36,451,800 | 703.8 |
| 2022年11月4日 | 699.7 | 708.9 | 699.1 | 702 | 42,728,700 | 702 |
| 2022年11月2日 | 702 | 707.4 | 700.6 | 702.3 | 45,086,600 | 702.3 |
| 2022年11月1日 | 705.4 | 707.2 | 701.2 | 703.3 | 34,022,800 | 703.3 |
| 2022年10月31日 | 694.4 | 700.5 | 692.3 | 699.5 | 39,451,900 | 699.5 |
| 2022年10月28日 | 697 | 699 | 687.5 | 687.5 | 54,071,200 | 687.5 |
| 2022年10月27日 | 709.5 | 710.1 | 691 | 691.1 | 44,516,600 | 691.1 |
| 2022年10月26日 | 711.5 | 716.5 | 709.4 | 709.4 | 43,590,800 | 709.4 |
| 2022年10月25日 | 712 | 717.8 | 709.4 | 710.1 | 52,191,300 | 710.1 |
| 2022年10月24日 | 705 | 707.7 | 699.4 | 705.3 | 52,455,000 | 705.3 |
| 2022年10月21日 | 688 | 698.2 | 687 | 695.3 | 40,312,100 | 695.3 |
| 2022年10月20日 | 687.9 | 694.4 | 686.6 | 693 | 43,652,500 | 693 |
| 2022年10月19日 | 681.9 | 688.7 | 680.1 | 688.7 | 56,380,500 | 688.7 |
| 2022年10月18日 | 684.9 | 688.7 | 679.4 | 681.9 | 67,082,000 | 681.9 |
| 2022年10月17日 | 669.2 | 681 | 668 | 675.3 | 99,461,500 | 675.3 |
| 2022年10月14日 | 658.9 | 663.9 | 655.4 | 659.2 | 64,104,200 | 659.2 |

図 1　(株) 三菱 UFJ フィナンシャル・グループのページ：2022 年 10 月 14 日から 11 月 11 日までの 20 日間の株価データを表示している．

採点の都合上，以下の要件を満たすこととする：

(1) StockPrice クラスは変更しない；
(2)「if __name__」内では，以下を行う；
　(a) コマンドライン引数にて，対象ファイルのパスを取得する；
　(b) read_html 関数を用いて，対象ファイル内の html ソースを読み込む；
　(c) 日付と株価関連の値を正規表現により抽出する．なお，株価関連の値（始値，高値，安値，終値，出来高，調整後終値）は，いずれも同じパターンで記述されている；
　(d) 抽出した日付と株価関連の値を属性にもつ StockPrice クラスのインスタンスを生成し，後述する StockPrices クラスに append する；
(3) StockPrices クラスを実装する；
　(a) list クラスを継承する；
　(b) StockPrice クラスのインスタンス以外を append できないようにする；
(4) 不要なモジュールは import しない；
(5) デバッグ用の print など必要ないものはコメントアウトし，出力が図 2 になるようにする；

```
PS C:\Users\admin\Desktop\ProA2> python lec08/kadai17/stock_prices.py lec08/data/ufj.html
StockPriceオブジェクト以外は追加できません
20日間の株価データ：
StockPrice(date='2022年11月11日', hajimene='715', takane='716.4', yasune='706.1', owarine='710.8', dekidaka='47,598,200', owarine2='710.8')
StockPrice(date='2022年11月10日', hajimene='707.3', takane='711.3', yasune='706.4', owarine='710.7', dekidaka='28,718,200', owarine2='710.7')
StockPrice(date='2022年11月9日', hajimene='710', takane='712.6', yasune='706.4', owarine='708.5', dekidaka='33,052,600', owarine2='708.5')
StockPrice(date='2022年11月8日', hajimene='707.3', takane='714', yasune='704.9', owarine='710.8', dekidaka='42,531,800', owarine2='710.8')
StockPrice(date='2022年11月7日', hajimene='708', takane='709', yasune='703.3', owarine='703.8', dekidaka='36,451,800', owarine2='703.8')
StockPrice(date='2022年11月4日', hajimene='699.7', takane='708.9', yasune='699.1', owarine='702', dekidaka='42,728,700', owarine2='702')
StockPrice(date='2022年11月2日', hajimene='702', takane='707.4', yasune='700.6', owarine='702.3', dekidaka='45,086,600', owarine2='702.3')
StockPrice(date='2022年11月1日', hajimene='705.4', takane='707.2', yasune='701.2', owarine='703.3', dekidaka='34,022,800', owarine2='703.3')
StockPrice(date='2022年10月31日', hajimene='694.4', takane='700.5', yasune='692.3', owarine='699.5', dekidaka='39,451,900', owarine2='699.5')
StockPrice(date='2022年10月28日', hajimene='697', takane='699', yasune='687.5', owarine='687.5', dekidaka='54,071,200', owarine2='687.5')
StockPrice(date='2022年10月27日', hajimene='709.5', takane='710.1', yasune='691', owarine='691.1', dekidaka='44,516,600', owarine2='691.1')
StockPrice(date='2022年10月26日', hajimene='711.5', takane='716.5', yasune='709.4', owarine='709.4', dekidaka='43,590,800', owarine2='709.4')
StockPrice(date='2022年10月25日', hajimene='712', takane='717.8', yasune='709.4', owarine='710.1', dekidaka='52,191,300', owarine2='710.1')
StockPrice(date='2022年10月24日', hajimene='705', takane='707.7', yasune='699.4', owarine='705.3', dekidaka='52,455,000', owarine2='705.3')
StockPrice(date='2022年10月21日', hajimene='688', takane='698.2', yasune='687', owarine='695.3', dekidaka='40,312,100', owarine2='695.3')
StockPrice(date='2022年10月20日', hajimene='687.9', takane='694.4', yasune='686.6', owarine='693', dekidaka='43,652,500', owarine2='693')
StockPrice(date='2022年10月19日', hajimene='681.9', takane='688.7', yasune='680.1', owarine='688.7', dekidaka='56,380,500', owarine2='688.7')
StockPrice(date='2022年10月18日', hajimene='684.9', takane='688.7', yasune='679.4', owarine='681.9', dekidaka='67,082,000', owarine2='681.9')
StockPrice(date='2022年10月17日', hajimene='669.2', takane='681', yasune='668', owarine='675.3', dekidaka='99,461,500', owarine2='675.3')
StockPrice(date='2022年10月14日', hajimene='658.9', takane='663.9', yasune='655.4', owarine='659.2', dekidaka='64,104,200', owarine2='659.2')
```

図 2　課題 17 の実行結果

リスト 1　stock_prices.py

```python
import sys
import re
from dataclasses import dataclass

@dataclass
class StockPrice:
    date: str
    hajimene: float
    takane: float
    yasune: float
    owarine: float
    dekidaka: int
    owarine2: float


class StockPrices(list):
    def append(self, sp:StockPrice):
        if not isinstance(sp, StockPrice):
            raise TypeError("StockPrice オブジェクト以外は追加できません")
        super().append(sp)


def read_html(file_path):
    with open(file_path, "r", encoding="utf8") as rfo:
        html = rfo.read()
    return html


if __name__ == "__main__":
    file_path = sys.argv[1]
    html = read_html(file_path).replace("\n", "")
    datepattern = r"<th scope=\"row\".*?>(\d{4}年\d{1,2}月\d{1,2}日)</th>"
    pricepattern = r"<td.*?><span.*?><span .*?><span .*?>([0-9.,]{1,})</span></span></span></td>"
    pattern = re.compile(datepattern+pricepattern*6)
```

```
35      sp_lst = StockPrices()
36      for match in pattern.findall(html):
37          sp_lst.append(StockPrice(*match))
38
39      try:
40          sp_lst.append(243)
41      except Exception as e:
42          print(e)
43
44      print(f"{len(sp_lst)}日間の株価データ：")
45      for sp in sp_lst:
46          print(sp)
```

**【課題 18：cs_teachers.py】**

配布した teachers.html から，正規表現を用いて CS 教員の名前，詳細ページの URL，職位，専門分野を抽出し，これらを属性に持つ Teacher オブジェクトを生成するコードを実装せよ．teachers.html は図 3 のソースコードである．


図 3　コンピュータサイエンス学部の教員紹介のページ：全 38 教員の情報が表示されている．

採点の都合上，以下の要件を満たすこととする：

(1)「if __name__」内では，以下を行う；
  (a) コマンドライン引数にて，対象ファイルのパスを取得する；
  (b) read_html 関数を用いて，対象ファイル内の html ソースを読み込む；
  (c) 正規表現により，CS 全教員（38 人）の情報（詳細ページの URL，名前，職位，専門分野）を抽出する；
  (d) ダミー教員であるほげほげ先生（URL：`hoge/fuga/:piyopiyo-1.html` ／職位：教授／専門分野：プログラミング）を追加する．ダミー教員を追加する意図は，Teacher クラスの属性に設定したディスクリプタが，ふさわしくない URL を検出できるかを検証するためである；
  (e) 例外処理により，ディスクリプタにより ValueError が raise されても異常終了することのないようにする；
  (f) コメントアウト部分は修正してもよいし，使用しなくてもよいが，出力結果は図 4 となるようにする；
(2) Teacher クラスを実装する；
  (a) 教員の名前，詳細ページの URL，職位，専門分野を属性に持つ；
  (b) URL を表す属性には，URL としてふさわしい文字列かどうかを正規表現により検証するディスクリプタを設置する；
  (c) 職位を表す属性には，「教授」，「准教授」，「講師」，「助教」のみを設定可能として，これらの職位かどうかを正規表現により検証するディスクリプタを設置する；
  (d) Teacher クラスのインスタンスを print した際に，図 4 のように出力されるように特殊メソッドをオーバーライドする；
(3) ディスクリプタクラスを *1* **つだけ**実装する；
  (a) ディスクリプタをインスタンス化する際に正規表現を引数として渡せるように，イニシャライザのパラメータを設定する；
  (b) URL と職位の値を属性に代入する際に正規表現による値の検証を行い，条件を満たさない場合は図 4 のような ValueError を raise する；
(4) 不要なモジュールは import しない；

```
PS C:\Users\admin\Desktop\ProA2> python lec08/kadai18/cs_teachers.py lec08/data/teachers.html
リストの長さ：39
1 大野 澄雄 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1660／職位：教授／専門分野：情報システム、音声情報処理、
2 青木 輝勝 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1732／職位：教授／専門分野：画像理解、画像処理、コンピュ
3 石畑 宏明 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1629／職位：教授／専門分野：コンピュータアーキテクチャ・
4 Reijer Grimbergen (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1661／職位：教授／専門分野：人工知能、認知科学、
5 佐藤 公則 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1734／職位：教授／専門分野：画像認識、画像計測，バイオメ
6 瀬之口 潤輔 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1736／職位：教授／専門分野：人工知能、機械学習モデルに
7 服部 聖彦 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1737／職位：教授／専門分野：群知能、群ロボット、自律分散
8 松下 宗一郎 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1662／職位：教授／専門分野：コンピューターエンターテイ
9 井上 亮文 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1688／職位：准教授／専門分野：ヒューマンコンピュータイン
10 岩下 志乃 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1685／職位：准教授／専門分野：感性情報処理，自然言語処理
11 長名 優子 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1663／職位：准教授／専門分野：ニューラルネットワーク、遺
12 菊池 眞之 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1643／職位：講師／専門分野：視覚情報処理、神経情報科学、
13 福西 広晃 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1731／職位：講師／専門分野：データサイエンス、バイオイン
14 伏見 卓恭 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1728／職位：講師／専門分野：複雑ネットワーク解析，ウェブ
15 松岡 丈平 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1730／職位：講師／専門分野：情報システム，デジタル画像処
16 山口 淳 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1698／職位：講師／専門分野：生産政策、オペレーションズ・マ
17 武 博 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1742／職位：助教／専門分野：人間情報学；情報ネットワーク；生
18 佐々木 亮平 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1743／職位：助教／専門分野：最適化，デジタル信号処理)
19 生野 壮一郎 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1631／職位：教授／専門分野：数値シミュレーション，ハイ
20 梅川　通久 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1741／職位：教授／専門分野：人材政策、人文社会情報学、
21 大石 邦夫 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1654／職位：教授／専門分野：信号処理、電子回路)
22 亀田 弘之 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1640／職位：教授／専門分野：思考と言語、言語情報学、機械
23 串田 高幸 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1733／職位：教授／専門分野：クラウド，分散コンピューティ
24 杉本 岩雄 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1682／職位：教授／専門分野：アロマセンシング・アロマサイ
25 竹田 昌弘 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1694／職位：教授／専門分野：情報通信ネットワーク導入によ
26 布田 裕一 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1727／職位：教授／専門分野：情報セキュリティ、暗号技術、
27 宇田 隆哉 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1633／職位：准教授／専門分野：ネットワークセキュリティ)
28 細野 繁 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1735／職位：准教授／専門分野：サービス工学、サービスコンピ
29 山元 進 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1632／職位：准教授／専門分野：電子物性、量子分子動力学、大
30 金光 永煥 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1628／職位：講師／専門分野：並列分散処理、タスクスケジュ
31 千葉 康生 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1634／職位：講師／専門分野：超局所解析、代数解析、ｅラー
32 塩野 康徳 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1746／職位：講師／専門分野：知能情報、数理情報、情報教育
33 董 然 (URL：https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1738／職位：助教／専門分野：認知科学、ヒューマンロボットイン
34 posが不正な値です：実験助手
35 posが不正な値です：実験助手
36 posが不正な値です：実験助手
37 posが不正な値です：実験助手
38 posが不正な値です：特任講師
39 urlが不正な値です：https://www.teu.ac.jphoge/fuga/:piyopiyo-1.html
```

図 4　課題 18 の実行結果：CS 教員 38 人にダミー教員 1 人を追加して，リストの長さは 39 となっている例.

リスト 2　cs_teachers.py

```
1  import sys
2  from urllib.request import urlopen
3  import re
4
5  class Varidator:
6      def __init__(self, pattern):
7          self.pattern = re.compile(pattern)
8
9      def __set_name__(self, obj, attrname):
10         self.attrname = attrname
11
12     def __get__(self, obj, objtype=None):
13         return obj.__dict__[self.attrname]
14
15     def __set__(self, obj, value):
16         if not self.pattern.fullmatch(value):
17             raise ValueError(f"{self.attrname}が不正な値です：{value}")
18         obj.__dict__[self.attrname] = value
19
20
21 class Teacher:
22     url = Varidator(r"^https?://[^/][\w\d/%#$&?()~_.=+-]+")
23     pos = Varidator(r"(教授|准教授|講師|助教)")
```

```
24
25      def __init__(self, name, url, pos, field):
26          self.name = name
27          self.url = url
28          self.pos = pos
29          self.field = field
30
31      def __repr__(self):
32          return f"{self.name} (URL：{self.url}／職位：{self.pos}／専門分野：{self.field})"
33
34
35  def read_html(file_path):
36      with open(file_path, "r", encoding="utf8") as rfo:
37          html = rfo.read()
38      return html
39
40
41  if __name__ == "__main__":
42      file_path = sys.argv[1]
43      html = read_html(file_path).replace("\n", "")
44      pattern = re.compile(r"<a href=\"(/info/lab/teacher/cs/index.html\?id=\d+).*?\">.*?<h2
            >(.*?)</h2>.*?<h4>(.*?)</h4>.*?<p>(.*?)</p>")
45      match_lst = pattern.findall(html)
46      match_lst.append(("hoge/fuga/:piyopiyo-1.html", "ほげほげ", "教授", "プログラミング"))
47      print(f"リストの長さ：{len(match_lst)}")
48      for i, match in enumerate(match_lst, 1):
49          try:
50              name = match[1]
51              url = "https://www.teu.ac.jp"+match[0]
52              pos = match[2].split()[0]
53              field= match[3]
54              t = Teacher(name, url, pos, field)
55              print(i, t)
56          except Exception as e:
57              print(i, e)
```