Praktiskais darbs programmēšanā

Armands Zūkers

```
1  from urllib.request import urlopen
2  from bs4 import BeautifulSoup
3  from urllib.parse import urlparse
4  from urllib.parse import parse_qs
5  import mysql.connector
6  import requests
7  import ftplib
8  import os
9  #Izveidojam savienojumu ar Datu Bazi
10 #Nepieciesama atlauja no servera puses konkretai IP adresei kas izplida
       so pieslegumu
11 cnx = mysql.connector.connect(user='kamlv_onlinestock', password='
       av160377mnt22',
12                                host='server1.firsthost.lv',
13                                database='kamlv_online_stock')
14 cursor = cnx.cursor()
15
16 #Izveidojam savienojumu ar FTP lai ieladetu bildes
17 ftp= ftplib.FTP('www.online-stock.lv')
18 ftp.login('onlinestock@online-stock.lv', 'av160377mnt22')
19
20 # Links kuru kidasim
21 html = urlopen("http://euroshops.lv/index.php?route=product/category&
       path=59_860825302&limit=50")
22
23 # Noradam ko mes isi meklejam pec tegiem
24 bsObj = BeautifulSoup(html)
25 recordList = bsObj.findAll("div", {"class": "product-layout product-
       list col-xs-12"})
26
27 # Taisam sarakstu
28 for record in recordList:
29      #Panemam datus
30      title = record.find("h4").get_text().strip()
31      price = record.find("p", {"class": "price"}).get_text().strip()
32      bilde = record.find('img').attrs['src']
33      links = record.find('a').attrs['href']
34      pandoc
35      #Pakidajam linku lai dabutu ara kategoriju un preces ID ar perseri
36      product_url = record.find('a').attrs['href']
37      parsed = urlparse(product_url)
38
39      #No linka izvelkam precesid
40      qs = parse_qs(parsed.query)['product_id'][0]
41
42      #No linka izvelkam kategoriju
43      qc = parse_qs(parsed.query)['path'][0]
44      res = qc.partition('_')[2]
45
46      #Dabjam bildes nosaukumu prieks DB
```

```
47        b = bilde.partition('http://euroshops.lv/image/cache/catalog/')[2]
48
49
50        #Parbaudam vai prece ar sadu ID ir datu baze
51        cursor.execute("select * from monte_products where productID='{}'"
              .format(qs))
52        row = cursor.fetchone()
53        if row == None:
54             #jA NAV PRECES AR SADU ID taisam jaunu preci
55             cursor.execute("insert into monte_products (productID,
                  categoryID, name_lv, picture1) VALUES
                  ('{}','{}','{}','{}')".format(qs,res,title,b))
56
57             #Liekam Cenu attiecigajai precei jo db cnas ir cita tabula(
                  Lai skripts stradatu html janomaina Eur simbols)
58             cursor.execute("insert into monte_prices (pid, price, enabled
                  , stock) VALUES ('{}','{}','1','10')".format(qs,price.
                  split('&euro;')[0]))
59
60             #Ladejam bildi lielo un mazo (vinas gan ir vienadas)
61             url = bilde
62             r = requests.get(url, allow_redirects=True)
63             open(b, 'wb').write(r.content)
64
65             ftp.cwd('/pictures/thumb/')
66             uploadfile= open(b, 'rb')
67             ftp.storbinary('STOR ' + b, uploadfile)
68             ftp.cwd('/pictures/big/')
69             uploadfile= open(b, 'rb')
70             ftp.storbinary('STOR ' + b, uploadfile)
71             uploadfile.close()
72             #izdzesam lokalo bildi
73             os.remove(b)
74             print("Prece Ieladeta: " + title)
75        else:
76             #ja IR prece ar sadu ID katram gadijumam updatojam Cenu (Lai
                  skripts stradatu html janomaina Eur simbols)
77             cursor.execute("update monte_prices set price='{}' where pid
                  ='{}'".format(price.split('&euro;')[0], qs))
78             print("Cena atjaunota: " + title)
79   print ("Finito")
```