CS3723 Pgm 6 Python (70 pts)

© 2018 Larry W. Clark, this document may not be copied to any other website.

This is a continuation of program #5. In this program, you will compare addresses to determine how close they match. This isn't just a simple comparison.

Source code:

* p6Driver.py will contain your driving software as was done in program #5 with modifications to invoke the comparison code. Copy p5Driver.py into p6Driver.py.
* address.py is needed in program#6.
* checkAddress.py is a new Python source file which will contain your source code that provides several functions including

**parseAddress** – determines the parts of the street lines and returns those and the rest of the address as a new dictionary (or object for extra credit). This method might raise an AddressSyntaxException.

***other functions*** for comparing addresses unless doing the extra credit

For each customer:

* Show the customer.
* In program #5, you retrieved multiple addresses for a customer. Each address was represented by a dictionary. You need to save each dictionary for a customer into one list.
* For each address:
  + Show the address with a sequence number. (This was done in program #5.)
  + Break the street line (which was concatenated from multiple LINE commands in program #5) into its parts (streetNr, streetName, streetType, direction, apartmentNumber). This part of the assignment requires *thinking*.
  + Show the parts of a street line unless an exception was raised.
  + Show pairs of addresses (referencing the sequence number) and provide a score (0..100) of how closely these match.

Your business people have provided this guidance:

1. Your code should recognize equivalence of common abbreviations. It is probably best to map them to the longer value.

Street types (not an exhaustive list):

* RD, ROAD
* AVE, AVENUE
* ST, STREET

Directions (not an exhaustive list):

* SOUTH, S (when appropriate)
* SW, SOUTHWEST, SOUTH WEST (when appropriate)

2. There is usually at least one word of a street name before a street type, assuming the street type exists. (e.g., 123 E ST)

3. The direction (e.g., SOUTH) might occur after the street number or after the street type.

4. The combined street lines must begin with a number.

5. There are a few (not many rules) for syntactically invalid addresses. If such an address is discovered, print a message about the address being invalid and ignore it in the scoring.

**Example addresses including name, street address, city, state, and zip**

BOB WIRE

1 123 DIRT RD

SAN ANTONIO, TX 78210

2 123 DIRT

SAN ANTONIO, TX 78210

3 123 DIRT LN

SAN ANTONIO, TX 78210

PENNY LOAFER

1 111 SHOE LN

SAN ANTONIO, TX 78249-1234

2 111 SHOE ST

SAN ANTONIO, TX 78249-1234

3 111 BOOT ST SOUTH APT 5A

SAN ANTONIO, TX 78230

4 111 S BOOT STREET APT 5A

SAN ANTONIO, TX 78230

FLO N WATER

1 45 S.W. VISTA RIO GRANDE RD

SAN ANTONIO, TX 78210

2 45 S WEST VISTA RIO GRANDE RD.

SAN ANTONIO, TX 78210

3 45 SOUTHWEST VISTA RIO GRANDE ROAD

SAN ANTONOI, TX 78210

4 SOUTHWEST VISTA RIO GRANDE ROAD

SAN ANTONIO, TX 78210

HOLLY WOOD

1 12 WEST AVE APT 23D

LOS ANGELES, CA 90009

2 12 WEST AVE APT 23D

LOS ANGELES, 90009

3 12 WEST A ST APT 23

LOS ANGELES, CA 90009

4 12 W. A ST APT 23

LOS VEGAS, CA 90009

BILL MELATER (in the extra credit file)

1 36A E COMMERCE ST NR 23

LOS ANGLES, CA 90009-2312

2 36-A COMMERCE ST EAST APT. NR. 5

LOS ANGELES, CA 90009

3 36A COMMERCE APT 5

LOS ANGELES, CA 9009

4 36A E COMMERCE ST #5

LOS ANGELES, CA 90009

5 36A EAST COMMEREC APT. # 5

LOS ANGELES, CA 90009

Scoring Matches:

|  |  |  |
| --- | --- | --- |
| Seq | Information | Scoring |
| 1 | Street number | +20 both not empty and they match  -20 both have values, but they don't match  0 both empty  -20 only one empty |
| 2 | Street Type | +10 both not empty and they match  -10 both have values, but they don't match  +10 both empty  +5 only one empty |
| 3 | Direction | +5 both not empty and they match  -10 both have values, but they don't match  +5 both empty  -5 only one empty |
| 4 | AptNum | +20 both not empty and they match  -20 both have values, but they don't match  +10 both empty  -10 only one empty  5 mostly match (scale) where both have values |
| 5 | City | +20 both not empty and they match  -20 both have values, but they don't match  +10 both empty  -10 only one empty  15 mostly match (scale) where both have values |
| 6 | State | +10 both not empty and they match  -20 both have values, but they don't match  0 both empty  0 only one empty |
| 7 | Street Name | +20 both not empty and they match  -5 both have values, but they don't match  -20 both empty  -20 only one empty  10 mostly match (scale) where both have values |
| 8 | Zip Code | +80 both not empty, both len of 10, and they match (if the ZIP+4 values match, these are most likely the same address)  +5 both not empty, both len of 5, and they match  +0 same len, but they don't match. Note that zip codes change a lot without all the addresses being updated  +5 lens not equal, the first 5 characters match  +0 lens not equal, the first 5 characters don't match. |

Notes:

1. If the computed score is less than 0, set it to 0. If it is greater than 100, set it to 100.

2. For scoring where **mostly match** is provided:

* Only use when the values are both not empty and they do not match.
* Use the SequenceMatcher's ratio to see if they **mostly match**.
* If the ratio is greater than 0.6, add the ratio \* the value in the table to the score.
* Otherwise, score using the both have values, but they do not match value.

3. Apartment numbers come in a variety of formats. Without extra credit, simply handle APT followed by white space and then the number (which isn't necessarily a number) (e.g., APT 5A). See extra credit for handling the other possibilities.

4. Some punctuation isn't valuable.

5. Your code should **raise** several **AddressSyntaxException** **exceptions** which should be caught by **parseAddress**. Each should be passed an appropriate error message. Example:

**raise AddressSyntaxException** ("Missing state")

To print it in a try … except:

except (*Exception1, Exception2, …*) as e:

print( "\*\*\*", str(e.args[1]), "\*\*\*")

The exception to be raised and corresponding possible messages are shown:

**AddressSyntaxException**

* + Missing address lines (i.e., no address LINE at all)
  + Missing city
  + Missing state
  + Missing zip
  + An address must begin with a number (e.g., 123, 36A)
  + Zip codes must be 5 digits followed by an optional "-" and an optional 4 digits.

**Extra Credit #1 (7 points) – using Python ParsedAddress object**

* **Late** submissions will **not** receive extra credit.
* Your **parseAddress** function should return a **ParsedAddress** object which has attributes for streetNum, aptNum, streetType, streetName, direction, city, state, and zip
* Instead of comparison functions in checkAddress.py, you should have a **ParsedAddress.py** file which will contain instance methods for those comparisons.
* Your code must properly handle all cases in p6Input.txt to be eligible for extra credit #1.

**Extra Credit #2 (200 / n) – handle the variety of apartment numbers**

* **Late** submissions will **not** receive extra credit.
* You must successfully complete **extra credit #1** to be eligible for **extra credit #2**.
* They may be preceded with APT, NR, and/or #. Frequently those have periods and run into the actual apartment number (which isn't necessarily a number).
* Your code must properly handle all cases in p6ExtraInput.txt to be eligible for extra credit #2.
* Some examples of equivalent apartment numbers:
  + APT #5A
  + NR 5A
  + #5-A
  + APT 5-A
  + NR. 5A
  + NR. 5-A
  + APT NR 5A
  + APT. NR. 5-A
  + #5A
  + APARTMENT #5A

Sample (partial) Output:

BOB WIRE StNum Direction AptNum StType StName

1 123 DIRT RD

SAN ANTONIO, TX 78210

123 ROAD DIRT

2 123 DIRT

SAN ANTONIO, TX 78210

123 DIRT

3 123 DIRT LN

SAN ANTONIO, TX 78210

123 LANE DIRT

Address Address Score

1 2 95

1 3 80

2 3 95

PENNY LOAFER StNum Direction AptNum StType StName

1 111 SHOE LN

SAN ANTONIO, TX 78249-1234

111 LANE SHOE

2 111 SHOE ST

SAN ANTONIO, TX 78249-1234

111 STREET SHOE

3 111 BOOT ST SOUTH APT 5A

SAN ANTONIO, TX 78230

111 SOUTH 5A STREET BOOT

4 111 S BOOT STREET APT 5A

SAN ANTONIO, TX 78230

111 SOUTH 5A STREET BOOT

Address Address Score

1 2 100

1 3 20

1 4 20

2 3 40

2 4 40

3 4 100

FLO N WATER StNum Direction AptNum StType StName

1 45 S.W. VISTA RIO GRANDE RD

SAN ANTONIO, TX 78210

45 SOUTHWEST ROAD VISTA RIO GRANDE

2 45 S WEST VISTA RIO GRANDE RD.

SAN ANTONIO, TX 78210

45 SOUTHWEST ROAD VISTA RIO GRANDE

3 45 SOUTHWEST VISTA RIO GRANDE ROAD

SAN ANTONOI, TX 78210

45 SOUTHWEST ROAD VISTA RIO GRANDE

4 SOUTHWEST VISTA RIO GRANDE ROAD

SAN ANTONIO, TX 78210

\*\*\* Missing street number \*\*\*

Address Address Score

1 2 100

1 3 93

2 3 93

HOLLY WOOD StNum Direction AptNum StType StName

1 12 WEST AVE APT 23D

LOS ANGELES, CA 90009

12 23D AVENUE WEST

2 12 WEST AVE APT 23D

LOS ANGELES, 90009

\*\*\* Missing state \*\*\*

3 12 WEST A ST APT 23

LOS ANGELES, CA 90009

12 WEST 23 STREET A

4 12 W. A ST APT 23

LOS VEGAS, CA 90009

12 WEST 23 STREET A

Address Address Score

1 3 39

1 4 0

3 4 70

BILL MELATER StNum Direction AptNum StType StName

1 36A E COMMERCE ST NR 23

LOS ANGLES, CA 90009-2312

36A EAST 23 STREET COMMERCE

2 36-A COMMERCE ST EAST APT. NR. 5

LOS ANGELES, CA 90009

36A EAST 5 STREET COMMERCE

3 36A COMMERCE APT 5

LOS ANGELES, CA 9009

\*\*\* Zip is invalid \*\*\*

4 36A E COMMERCE ST #5

LOS ANGELES, CA 90009

36A EAST 5 STREET COMMERCE

5 36A EAST COMMEREC APT. # 5

LOS ANGELES, CA 90009

36A EAST 5 COMMEREC

Address Address Score

1 2 64

1 4 64

1 5 48

2 4 100

2 5 93

4 5 93