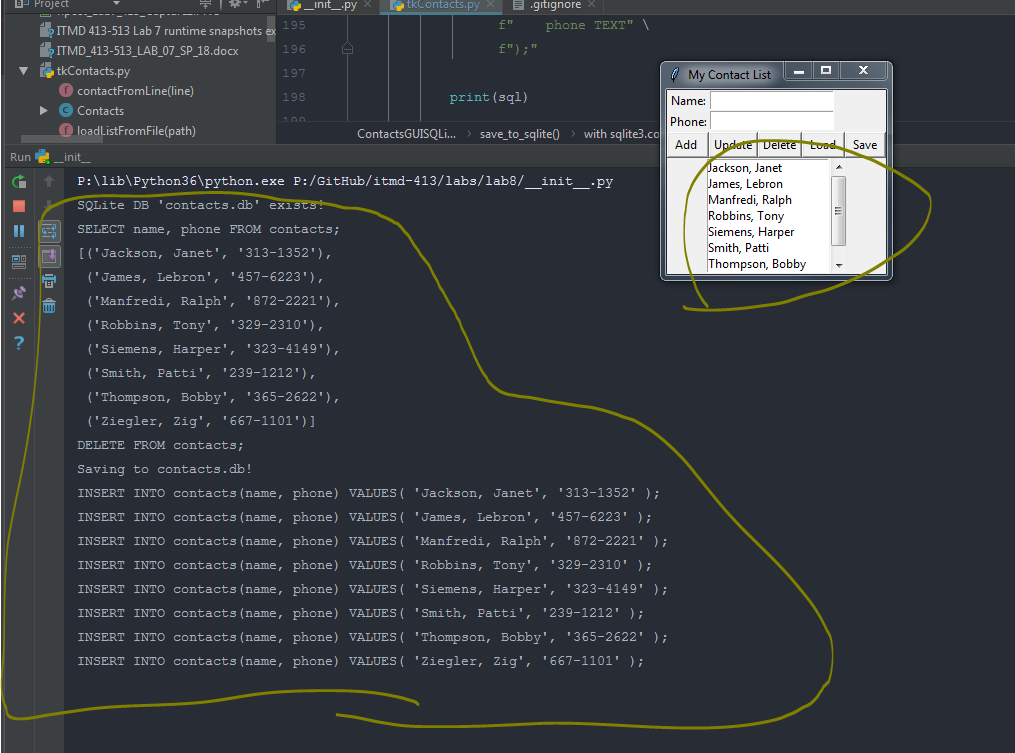
Henry Post

ITMD413

Lab 8, SQLite3 GUI



tkContacts.py

1. **import** os
2. **import** sqlite3
3. **from** pprint **import** pprint
4. **from** tkinter **import** \*

7. **class** Contact:
8. @staticmethod
9. **def** from\_line(line: str):
10. lines = line.split(',')
12. name = ','.join(lines[:-1])  # all but last
13. phone = lines[-1]  # last
15. contact = Contact(name, phone)
16. **return** contact
18. **def** as\_sqlite\_values(self):
19. **return** rf''''' '{self.name}', '{self.phone}' '''
21. **def** \_\_init\_\_(self, name: str, phone: str):
22. name = name.strip()
23. phone = phone.strip()
25. self.name: str = name
26. self.phone: str = phone
28. **def** \_\_str\_\_(self):
29. **return** f"{self.name}, {self.phone}"
31. **def** \_\_lt\_\_(self, other):
32. **return** self.name < other.name
34. **def** \_\_gt\_\_(self, other):
35. **return** self.name > other.name
37. **def** \_\_iter\_\_(self):
38. **yield** self.name
39. **yield** self.phone

42. default\_contact\_list = [
43. ('Siemens, Harper', '323-4149'),
44. ('Smith, Patti', '239-1212'),
45. ('Jackson, Janet', '313-1352'),
46. ('Manfredi, Ralph', '872-2221'),
47. ('Thompson, Bobby', '365-2622'),
48. ('James, Lebron', '457-6223'),
49. ('Ziegler, Zig', '667-1101'),
50. ('Robbins, Tony', '329-2310')
51. ]
53. default\_contact\_list = [Contact(name, phone) **for** (name, phone) **in** default\_contact\_list]

56. **class** ContactsGUI:
57. **def** selection(self):
58. **print**("At %s of %d" % (self.select.curselection(), len(self.contacts)))
59. **return** int(self.select.curselection()[0])
61. **def** get\_name(self):
62. **return** self.nameVar.get()
64. **def** get\_phone(self):
65. **return** self.phoneVar.get()
67. **def** get\_contact\_from\_fields(self):
68. **return** Contact(self.get\_name(), self.get\_phone())
70. **def** get\_selected(self):
71. **return** self.contacts[self.selection()]
73. **def** add\_contact(self):
74. self.contacts.append(self.get\_contact\_from\_fields())
75. self.set\_list()
77. **def** update\_selected\_contact(self):
78. self.contacts[self.selection()] = self.get\_contact\_from\_fields()
79. self.set\_list()
81. **def** delete\_selected\_contact(self):
82. **del** self.contacts[self.selection()]
83. self.set\_list()
85. **def** load\_selected\_contact(self):
86. name, phone = self.contacts[self.selection()]
87. self.nameVar.set(name)
88. self.phoneVar.set(phone)
90. **def** \_\_init\_\_(self, path):
91. self.path = path
92. self.contacts = []
93. self.load\_from\_disk()
95. self.root = Tk()
97. self.root.winfo\_toplevel().title("My Contact List")
99. self.framebuttons = Frame(self.root)
100. self.framebuttons.pack(fill=BOTH, expand=YES)
102. Label(self.framebuttons, text="Name:").grid(row=0, column=0, sticky=N)
103. self.nameVar = StringVar()
104. self.name = Entry(self.framebuttons, textvariable=self.nameVar)
105. self.name.grid(row=0, column=1, sticky=W)
107. Label(self.framebuttons, text="Phone:").grid(row=1, column=0, sticky=N)
108. self.phoneVar = StringVar()
109. self.phone = Entry(self.framebuttons, textvariable=self.phoneVar)
110. self.phone.grid(row=1, column=1, sticky=N)
112. self.framebuttons = Frame(self.root)  # add a row of buttons
113. self.framebuttons.pack()
114. self.btn\_add = Button(self.framebuttons, text=" Add  ", command=self.add\_contact)
115. self.btn\_update = Button(self.framebuttons, text="Update", command=self.update\_selected\_contact)
116. self.btn\_delete = Button(self.framebuttons, text="Delete", command=self.delete\_selected\_contact)
117. self.btn\_load = Button(self.framebuttons, text=" Load ", command=self.load\_selected\_contact)
118. self.btn\_save = Button(self.framebuttons, text=" Save ", command=self.set\_list)
120. self.btn\_add.pack(side=LEFT)
121. self.btn\_update.pack(side=LEFT)
122. self.btn\_delete.pack(side=LEFT)
123. self.btn\_load.pack(side=LEFT)
124. self.btn\_save.pack(side=LEFT)
126. self.framebuttons = Frame(self.root)  # allow for selection of names
127. self.framebuttons.pack()
128. self.scroll = Scrollbar(self.framebuttons, orient=VERTICAL)
129. self.select = Listbox(self.framebuttons, yscrollcommand=self.scroll.set, height=7)
130. self.scroll.config(command=self.select.yview)
131. self.scroll.pack(side=RIGHT, fill=Y)
132. self.select.pack(side=LEFT, fill=BOTH)
134. self.set\_list()
136. **def** set\_list(self):
137. """Resets displayed list from self.contacts."""
138. self.contacts.sort()
139. self.select.delete(0, END)
141. **for** contact **in** self.contacts:
142. self.select.insert(END, contact.name)
144. self.save\_to\_disk()
146. **def** load\_default\_contacts(self):
147. self.contacts = default\_contact\_list
149. **def** create\_blank\_file(self):
150. with open(self.path, 'w') as f:
151. **pass**
153. **def** create\_default\_file(self):
154. self.load\_default\_contacts()
155. self.save\_to\_disk()
157. **def** save\_to\_disk(self):
158. self.save\_to\_file()
160. **def** load\_from\_disk(self):
161. self.load\_from\_file()
163. **def** load\_from\_file(self):
164. self.contacts = ContactsGUI.load\_from\_file\_s(self.path)
166. @staticmethod
167. **def** load\_from\_file\_s(path) -> list:
168. ret = []
169. with open(path, 'r') as f:
170. **for** line **in** f:
171. line = line.replace('\r', '').replace('\n', '')
172. contact = Contact.from\_line(line)
173. ret.append(contact)
174. **return** ret
176. **def** save\_to\_file(self):
177. with open(self.path, 'w') as f:
178. **print**("Writing list to file at '" + self.path + "'.")
180. **for** contact **in** self.contacts:
181. line = str(contact)
182. f.write(line + '\n')

185. **class** ContactsGUISQLite(ContactsGUI):
186. contacts\_table\_name = 'contacts'
188. **def** create\_default\_tables(self):
189. with sqlite3.connect(self.path) as conn:
190. c = conn.cursor()
192. sql = f"CREATE TABLE {self.contacts\_table\_name} (" \
193. f"    contact\_id INTEGER PRIMARY KEY AUTOINCREMENT," \
194. f"    name TEXT," \
195. f"    phone TEXT" \
196. f");"
198. **print**(sql)
200. c.execute(sql)
202. **def** save\_to\_sqlite(self):
204. with sqlite3.connect(self.path) as conn:
205. c = conn.cursor()
207. sql = f"DELETE FROM {self.contacts\_table\_name};"
208. **print**(sql)
209. c.execute(sql)  # delete all
211. **print**(f"Saving to {self.path}!")
213. **for** contact **in** self.contacts:
214. contact: Contact
215. sql = f'INSERT INTO {self.contacts\_table\_name}(name, phone) VALUES(' + \
216. contact.as\_sqlite\_values() + \
217. f');'
219. **print**(sql)
221. c.execute(sql)
223. **def** load\_from\_sqlite(self):
224. path = os.path.abspath(self.path)
226. **if** **not** os.path.isfile(path):  # create blank file if file DNE
227. **print**(f"File '{self.path}' DNE. Creating blank file.")
229. self.create\_blank\_file()
230. self.create\_default\_tables()
231. self.load\_default\_contacts()
232. self.save\_to\_disk()
233. **return**
235. **print**(f"SQLite DB '{self.path}' exists!")
237. with sqlite3.connect(self.path) as conn:
238. c = conn.cursor()
240. sql = f"SELECT name, phone FROM {self.contacts\_table\_name};"
241. **print**(sql)
242. c.execute(sql)
244. results = c.fetchall()
245. pprint(results)
247. **for** result **in** results:
248. contact = Contact(result[0], result[1])
249. self.contacts.append(contact)

252. **def** save\_to\_disk(self):
253. self.save\_to\_sqlite()
255. **def** load\_from\_disk(self):
256. self.load\_from\_sqlite()

\_\_init\_\_.py

1. **import** os
3. **from** labs.lab8.tkContacts **import** ContactsGUI, Contact, ContactsGUISQLite
5. contacts\_file = 'contacts.db'
7. **if** \_\_name\_\_ == '\_\_main\_\_':
8. contacts\_gui = ContactsGUISQLite(contacts\_file)
10. contacts\_gui.root.mainloop()