

Deborah Barndt

ITMD 513 Open Source Programming

Professor Dr. Sam

Final

4-29-19

1 '''

2 Deborah Barndt

3 4-29-19

4 HospitalStay.py

5 final: Patient Fees - using Graphical User Interface (GUI)

6

7 This program will create menu that allows the user to enter a type of surgery

8 and a type of medication, and check the patient out of the hospital. When the

9 patient checks out, the total charges will be displayed in a window format. The

10 GUI program will compute a patient's bill for a hospital stay. The different

11 components of the program are: the PatientAccount class, the Surgery class, and

12 the Pharmacy class.

13

14 The PatientAccount class will keep a total of the patient's charges, and will

15 also keep track of the number of days spent in the hospital. The group must

16 decide on the hospital's daily rate. The Surgery class will have stored within

17 it the charges for at least five types of surgery, and it can update the

18 charges variable of the PatientAccount class. The Pharmacy class will have

19 stored within it the price of at least five type of medication, and it can

20 update the charges variable PatientAccount class.

21

22 Written by Deborah Barndt.

23 '''

```
24
25 from tkinter import *
26 from tkinter import ttk
27 from tkinter import messagebox
28 import re
29 from datetime import datetime
30 from datetime import date
31
32 # PatientAccount class that will keep a total of the patient's charges.
33 class PatientAccount:
34     # Constructor function for the patient's information such as their first name,
35     # last name, date of birth, hospital daily rate, days in the hospital, and the
36     # hospital charges.
37     def __init__(self):
38         self.__patientFirstName = None
39         self.__patientLastName = None
40         self.__patientDOB = None
41         self.__hospDailyRate = 3949.95
42         self.__numDaysInHosp = 0
43         self.__hospCharges = {}
44
45     # Function to get the patient's first name.
46     def getPatientFirstName(self):
47         return self.__patientFirstName
48
49     # Function to set the patient's first name.
50     def setPatientFirstName(self, patientFirstName):
51         self.__patientFirstName = patientFirstName
52
```

```
53     # Function to get the patient's last name.
54     def getPatientLastName(self):
55         return self.__patientLastName
56
57     # Function to set the patient's last name.
58     def setPatientLastName(self, patientLastName):
59         self.__patientLastName = patientLastName
60
61     # Function to get the patient's date of birth (DOB).
62     def getPatientDOB(self):
63         return self.__patientDOB
64
65     # Function to set the patient's date of birth (DOB).
66     def setPatientDOB(self, patientDOB):
67         self.__patientDOB = patientDOB
68
69     # Function to get the hospital's daily rate.
70     def getHospDailyRate(self):
71         return self.__hospDailyRate
72
73     # Function to set the hospital's daily rate.
74     def setHospDailyRate(self, dailyRate):
75         self.__hospDailyRate = dailyRate
76
77     # Function to get the number of days the patient has been in the hospital.
78     def getNumDaysInHosp(self):
79         return self.__numDaysInHosp
80
81     # Function to set the number of days the patient has been in the hospital.
```

```
82     def setNumDaysInHosp(self, numDays):
83         self.__numDaysInHosp = numDays
84
85     # Function to get the patient's charges for their hospital stay.
86     def getHospCharges(self):
87         return self.__hospCharges
88
89     # Function to set the patient's charges for their hospital stay.
90     def setHospCharges(self, dailyRate, numDays, proCost, medCost):
91         self.__hospTotal = (dailyRate * numDays) + proCost + medCost
92         self.__hospCharges = self.__hospTotal
93
94     # Surgery class that will have stored the charges for at least five types of surgery.
95     class Surgery:
96         # Constructor function for the types of surgery procedures and prices.
97         def __init__(self):
98             self.__procedures = {
99                 'Appendectomy': 1500,
100                 'Breast Biopsy': 500,
101                 'Coronary Artery Bypass': 40000,
102                 'Hip Replacement': 39299,
103                 'Knee Replacement': 49500,
104                 'Tonsillectomy': 8500,
105                 'Thyroidectomy': 10824
106             }
107
108         self.__procedures = None
109         self.__price = 0
110
```

```
111     # Function to get the surgery procedures for the patient.
112     def getProcedures(self):
113         return self.__procedures
114
115     # Function to set the surgery procedures for the patient.
116     def setProcedures(self, procedures):
117         self.__procedures = procedures
118
119     # Pharmacy class that will have stored the price of at least five types of medication.
120     class Pharmacy:
121         #Constructor function for the types of medications and prices.
122         def __init__(self):
123             self.__medications = {
124                 'Amoxil': 16.94,
125                 'Deltasone': 113,
126                 'Levoxyl': 81,
127                 'Lipitor': 165,
128                 'Neurontin': 241,
129                 'Vicodin': 114.85,
130                 'Zestril': 1257
131             }
132
133     # Function to get the medications for the patient.
134     def getMedications(self):
135         return self.__medications
136
137     # Function to set the medications for the patient.
138     def setMedications(self, medications):
139         self.__medications = medications
```

```
140
141 # Function to contain the surgery objects.
142 def surgeryCost(surgery, newPrice):
143     procedure = [
144         'Appendectomy',
145         'Breast Biopsy',
146         'Coronary Artery Bypass',
147         'Hip Replacement',
148         'Knee Replacement',
149         'Tonsillectomy',
150         'Thyroidectomy'
151     ]
152
153     price = [1500, 500, 40000, 39299, 49500, 8500, 10824]
154
155     if(surgery == procedure[0]):
156         newPrice = price[0]
157
158     elif(surgery == procedure[1]):
159         newPrice = price[1]
160
161     elif(surgery == procedure[2]):
162         newPrice = price[2]
163
164     elif(surgery == procedure[3]):
165         newPrice = price[3]
166
167     elif(surgery == procedure[4]):
168         newPrice = price[4]
```

```
169
170     elif(surgery == procedure[5]):
171         newPrice = price[5]
172
173     elif(surgery == procedure[6]):
174         newPrice == price[6]
175
176     return newPrice
177
178 # Function to contain the pharmacy objects.
179 def pharmacyCost(pharmacy, newMedPrice):
180     medications = [
181         'Amoxil',
182         'Deltasone',
183         'Levoxyl',
184         'Lipitor',
185         'Neurontin',
186         'Vicodin',
187         'Zestril'
188     ]
189
190     price = [16.94, 113, 81, 165, 241, 114.85, 1257]
191
192     if(pharmacy == medications[0]):
193         newMedPrice = price[0]
194
195     elif(pharmacy == medications[1]):
196         newMedPrice = price[1]
197
```

```
198     elif(pharmacy == medications[2]):
199         newMedPrice = price[2]
200
201     elif pharmacy == medications[3]:
202         newMedPrice = price[3]
203
204     elif pharmacy == medications[4]:
205         newMedPrice = price[4]
206
207     elif pharmacy == medications[5]:
208         newMedPrice = price[5]
209
210     elif pharmacy == medications[6]:
211         newMedPrice = price[6]
212
213     return newMedPrice
214
215 # Function to calculate the hospital charges.
216 def calcCharge():
217     newPrice = 0
218     newMedPrice = 0
219
220     patient = PatientAccount()
221     patientsurgery = Surgery()
222     patientpharmacy = Pharmacy()
223     firstname = str(fNameText.get(1.0, END))
224     lastname = str(lNameText.get(1.0, END))
225     #dob = datetime.date(dobText.get(1.0, END))
226     dob = str(dobText.get(1.0, END))
```



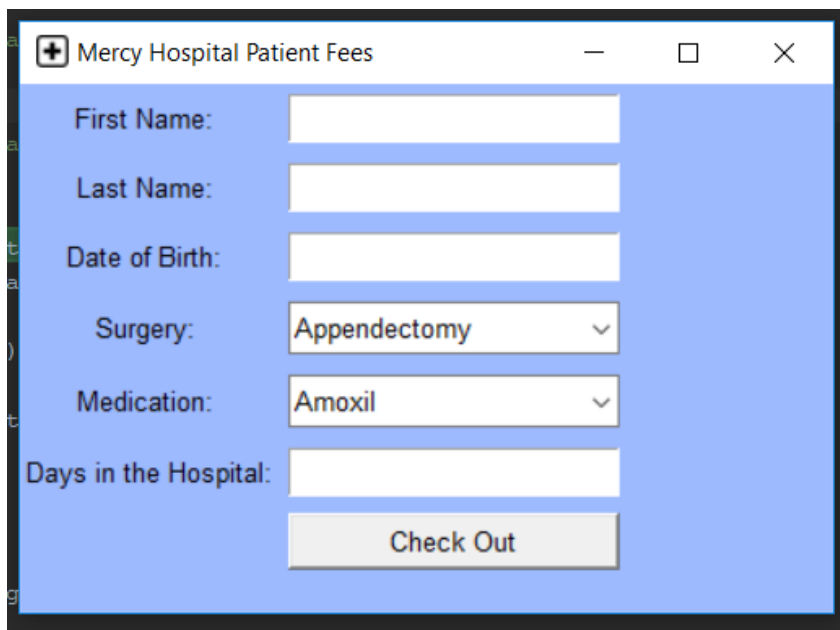
```
227
228     try:
229         numDays = float(daysText.get(1.0, END))
230
231     except ValueError as e:
232         messagebox.showerror("Error", "Number of days in hospital must be a number.")
233     return
234
235     """try:
236         datetime.strptime(dob, "%m/%d/%Y")
237
238     except ValueError as err:
239         messagebox.showerror("Error", "Date of birth format must be MM/DD/YYYY.")
240     return
241 """
242     # Check the validation of the user input.
243     if(firstname.strip() == ""):
244         messagebox.showerror('Error', 'First name cannot be empty.')
245
246     if(lastname.strip() == ""):
247         messagebox.showerror('Error', 'Last name cannot be empty.')
248
249     if(dob.strip() == ""):
250         messagebox.showerror('Error', 'Date of birth cannot be empty.')
251
252     if(numDays < 0):
253         messagebox.showerror('Error', 'Cannot enter negative amount of days.')
254
255     else:
```

```
256     patient.setPatientFirstName(firstname)
257     patient.setPatientLastName(lastname)
258     patient.setPatientDOB(dob)
259     patient.setNumDaysInHosp(numDays)
260
261     dailyRate = float('{0:2f}'.format(patient.getHospDailyRate()))
262
263     totalRate = dailyRate * numDays
264
265     surgery = str(surgeryList.get())
266     patientsurgery.setProcedures(surgery)
267
268     pharmacy = str(pharmacyList.get())
269     patientpharmacy.setMedications(pharmacy)
270
271     proCost = float('{0:2f}'.format(surgeryCost(surgery, newPrice)))
272     medCost = float('{0:2f}'.format(pharmacyCost(pharmacy, newMedPrice)))
273
274     patient.setHospCharges(dailyRate, numDays, proCost, medCost)
275     totalBill = float('{0:2f}'.format(patient.getHospCharges()))
276
277     # Display area for patient checkout and display in a pop-up messagebox.
278     messagebox.showinfo('Mercy Hospital Total Bill', 'First Name: ' +
279         patient.getPatientFirstName() + 'Last Name: ' +
280         patient.getPatientLastName() + 'Date of Birth: ' +
281         patient.getPatientDOB() + 'Rate: $' +
282         str('{0:2f}'.format(totalRate)) + '\n' + patientsurgery.getProcedures() + '$' +
283         str('{0:2f}'.format(proCost)) + '\n' + patientpharmacy.getMedications() + '$' +
284         str('{0:2f}'.format(medCost)) + '\nTotal Bill: $' + str('{0:2f}'.format(totalBill)))
```

```
285
286
287 # Create an instance for window to display.
288 window = Tk()
289
290 # Create the GUI window title, dimensions, and color.
291 window.title('Mercy Hospital Patient Fees')
292 window.geometry('400x260')
293 window.config(background='#9DBAFF')
294 window.iconbitmap('hospitalcross.ico')
295 window.option_add('*Dialog.msg.font', 'Franklin Book 16')
296
297 # Labels for the Patient Account information.
298 fNameLabel = Label(window, text = 'First Name: ', background='#9DBAFF', font=('Franklin Book', 10))
299 fNameLabel.grid(row = 0, column = 0, sticky = W+E)
300
301 lNameLabel = Label(window, text = 'Last Name: ', background='#9DBAFF', font=('Franklin Book', 10))
302 lNameLabel.grid(row = 1, column = 0, sticky = W+E)
303
304 dobLabel = Label(window, text = 'Date of Birth: ', background='#9DBAFF', font=('Franklin Book', 10))
305 dobLabel.grid(row = 5, column = 0, sticky = W+E)
306
307 surgeryLabel = Label(window, text = 'Surgery: ', background='#9DBAFF', font=('Franklin Book', 10))
308 surgeryLabel.grid(row = 6, column = 0, sticky = W+E)
309
310 pharmacyLabel = Label(window, text = 'Medication: ', background='#9DBAFF', font=('Franklin Book', 10))
311 pharmacyLabel.grid(row = 7, column = 0, sticky = W+E)
312
```

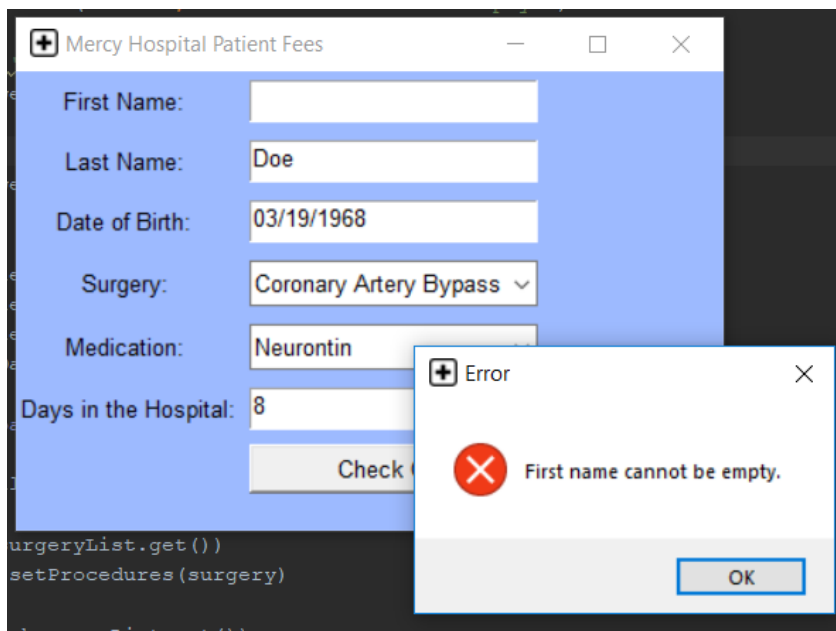
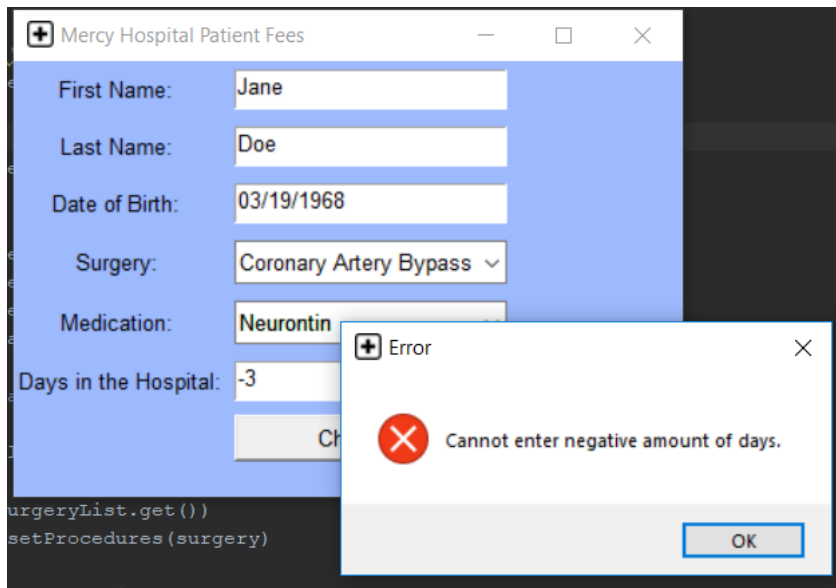
```
313 daysLabel = Label(window, text = 'Days in the Hospital:', background='#9DBAFF', font=('Franklin Book',
314 10))
315 daysLabel.grid(row = 8, column = 0, sticky = W+E)
316
317 # Text fields for the user input.
318 fNameText = Text(window, width = 15, height = 1, font=('Franklin Book', 10))
319 fNameText.grid(row = 0, column = 1, padx = 5, pady = 5, ipady = 2, sticky = W+E)
320
321 lNameText = Text(window, width = 15, height = 1, font=('Franklin Book', 10))
322 lNameText.grid(row = 1, column = 1, padx = 5, pady = 5, ipady = 2, sticky = W+E)
323
324 dobText = Text(window, width = 15, height = 1, font=('Franklin Book', 10))
325 dobText.grid(row = 5, column = 1, padx = 5, pady = 5, ipady = 2, sticky = W+E)
326
327 daysText = Text(window, width = 15, height = 1, font=('Franklin Book', 10))
328 daysText.grid(row = 8, column = 1, padx = 5, pady = 5, ipady = 2, sticky = W+E)
329
330 # Drop down menu for the list of surgeries.
331 surgeryChoice = StringVar()
332 surgeryList = ttk.Combobox(window, textvariable = surgeryChoice, font=('Franklin Book', 10))
333 surgeryList['values'] = ('Appendectomy', 'Breast Biopsy', 'Coronary Artery Bypass', 'Hip Replacement',
334                          'Knee Replacement', 'Tonsillectomy', 'Thyroidectomy')
335 surgeryList.current(0)
336 surgeryList.grid(row = 6, column = 1, padx = 5, pady = 5, ipady = 2, sticky = W)
337
338 # Drop down menu for the list medications.
339 pharmacyChoice = StringVar()
340 pharmacyList = ttk.Combobox(window, textvariable = pharmacyChoice, font=('Franklin Book', 10))
341 pharmacyList['values'] = ('Amoxil', 'Deltasone', 'Levoxyl', 'Lipitor',
```

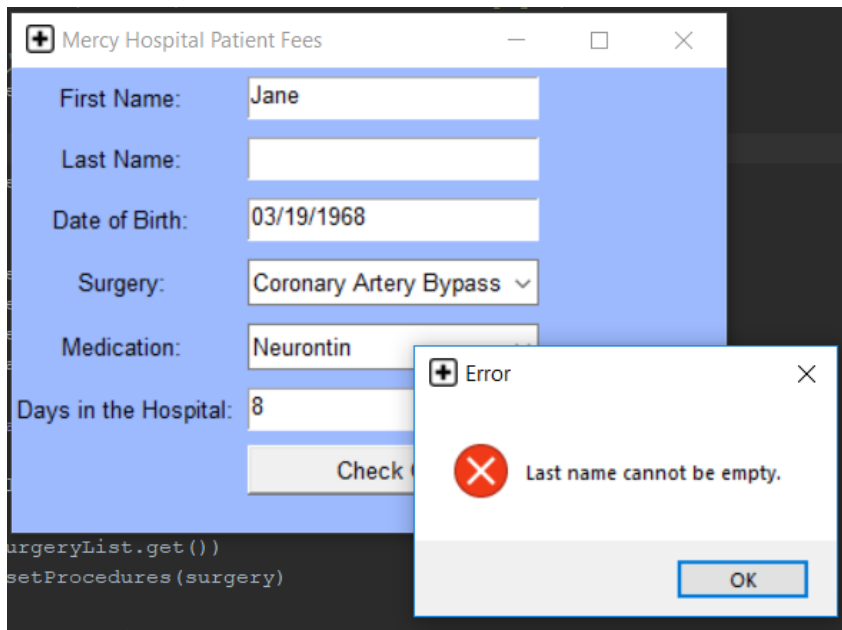
```
342         'Neurontin', 'Vicodin', 'Zestril')
343     pharmacyList.current(0)
344     pharmacyList.grid(row = 7, column = 1, padx = 5, pady = 5, ipady = 2, sticky = W)
345
346     # Checkout button to check out patient from the hospital.
347     calcBtn = Button(window, text = 'Check Out', font=('Franklin Book', 10), width = 10, command =
348     calcCharge)
349     calcBtn.grid(row = 9, column = 1, padx = 5, pady = 3, sticky = W+E)
350
351     # Create an event loop to display the GUI.
352     window.mainloop()
353
354     Output Result:
```



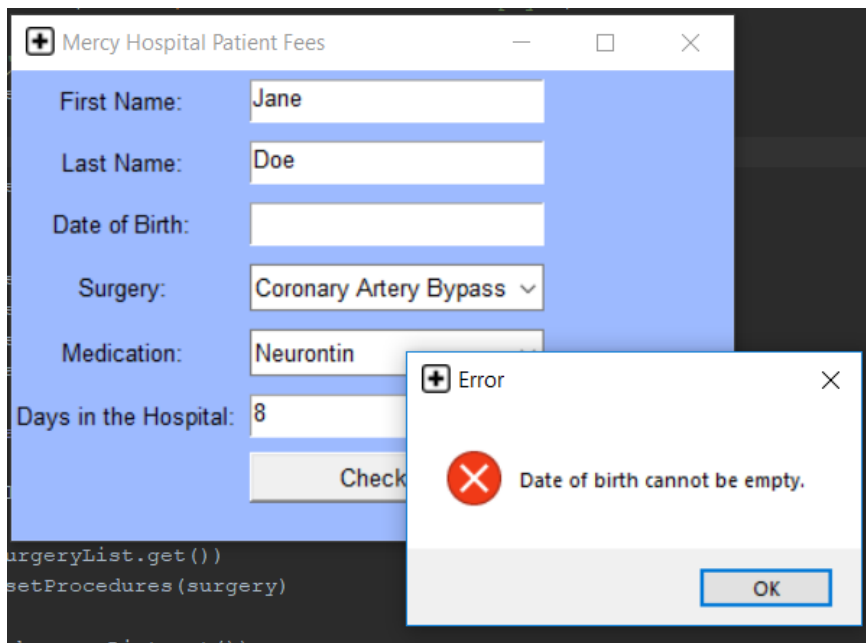
The screenshot shows a Tkinter window titled "Mercy Hospital Patient Fees". The window has a light blue background and contains several input fields and a button. The fields are labeled "First Name:", "Last Name:", "Date of Birth:", "Surgery:", "Medication:", and "Days in the Hospital:". The "Surgery:" field has a dropdown menu with "Appendectomy" selected. The "Medication:" field has a dropdown menu with "Amoxil" selected. The "Days in the Hospital:" field is empty. Below the "Days in the Hospital:" field is a "Check Out" button.

355





358



359

360

+

Mercy Hospital Patient Fees

First Name: Jane

Last Name: Doe

Date of Birth: 03/19/1975

Surgery: Coronary Artery Bypass

Medication: Vicodin

Days in the Hospital: 8

Check

+

Mercy Hospital Total Bill

i

First Name: Jane
Last Name: Doe
Date of Birth: 03/19/1975
Rate: \$31599.60
Coronary Artery Bypass \$40000.00
Vicodin \$114.85
Total Bill: \$71714.45

OK

```
surgeryList.get()
setProcedures(surgery)
```

361

+

Mercy Hospital Patient Fees

First Name: John

Last Name: Smith

Date of Birth: 01/27/1969

Surgery: Knee Replacement

Medication: Lipitor

Days in the Hospital: 4

Check

+

Mercy Hospital Total Bill

i

First Name: John
Last Name: Smith
Date of Birth: 01/27/1969
Rate: \$15799.80
Knee Replacement \$49500.00
Lipitor \$165.00
Total Bill: \$65464.80

OK

```
surgeryList.get()
setProcedures(surgery)
```


Mercy Hospital Patient Fees

First Name:

Last Name:

Date of Birth:

Surgery:

Medication:

Days in the Hospital:

Mercy Hospital Total Bill

i First Name: Lee
Last Name: Thompson
Date of Birth: 07/19/1974
Rate: \$23699.70
Hip Replacement \$39299.00
Levoxyl \$81.00
Total Bill: \$63079.70

362

Mercy Hospital Patient Fees

First Name:

Last Name:

Date of Birth:

Surgery:

Medication:

Days in the Hospital:

Mercy Hospital Total Bill

i First Name: Tina
Last Name: Lewis
Date of Birth: 11/27/1989
Rate: \$7899.90
Tonsillectomy \$8500.00
Amoxil \$16.94
Total Bill: \$16416.84

363