**Міністерство освіти та науки України  
Національний технічний університет України  
«Київський політехнічний інститут»  
Факультет прикладної математики  
Кафедра системного програмування і спеціалізованих  
комп’ютерних систем**

**Лабораторна робота №1**з дисципліни

**«Організація баз даних»**

**Варіант №11**

Виконав: Левунець Богдан Володимирович

Студент групи КВ-42

Перевірив(ла)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**м. Київ**

**2016**

**Структура бази даних у вигляді колекцій**

**Предметна галузь: Тип товару — товар;**

**Функція пошуку: Вивести типи товарів, де мінімальна ціна більше 10 грн**

**Текст програми**

**# model.py**

**import pickle**

*# checking, if pickle's file already exist*

**import** os.path

**class** ArticleType(object):

**def** \_\_init\_\_(self, name, country, overall\_vol, t\_num):

self.name = name

self.country = country

self.overall\_vol = overall\_vol

self.t\_num = t\_num

**def** get\_name(self):

**return** self.name

**def** get\_t\_num(self):

**return** self.t\_num

**def** get\_country(self):

**return** self.country

**def** get\_vol(self):

**return** self.overall\_vol

**def** set\_name(self, name):

self.name = name

**def** set\_country(self, country):

self.country = country

**def** set\_overall\_vol(self, vol):

self.overall\_vol = vol

**class** Article(object):

**def** \_\_init\_\_(self, name, amount, price, t\_num):

self.name = name

self.amount = amount

self.price = price

self.t\_num = t\_num

**def** get\_name(self):

**return** self.name

**def** get\_amount(self):

**return** self.amount

**def** get\_price(self):

**return** self.price

**def** get\_t\_num(self):

**return** self.t\_num

**def** set\_name(self, name):

self.name = name

**def** set\_amount(self, amount):

self.amount = amount

**def** set\_price(self, price):

self.price = price

**class** Database(object):

type\_list = []

art\_list = []

**def** \_\_init\_\_(self, t\_file, a\_file):

self.t\_file = t\_file

self.a\_file = a\_file

**if** os.path.exists(t\_file) **and** os.path.exists(a\_file):

self.pckl\_type = open(t\_file, **"rb"**)

self.pckl\_art = open(a\_file, **"rb"**)

self.type\_list = pickle.load(self.pckl\_type)

self.art\_list = pickle.load(self.pckl\_art)

self.pckl\_type.close()

self.pckl\_art.close()

**def** get\_type\_list(self):

**return** self.type\_list

**def** get\_art\_list(self):

**return** self.art\_list

**def** find\_type(self, name):

*"""return articles\_type if it's found or False otherwise"""*

**for** item **in** self.type\_list:

**if** item.get\_name() == name:

**return** item

**return** False

**def** find\_article(self, name):

*"""Return article if it's found or False otherwise"""*

**for** item **in** self.art\_list:

**if** item.get\_name() == name:

**return** item

**return** False

**def** del\_articles\_by\_t\_num(self, t\_num):

ind = 0

**while** ind < len(self.art\_list):

**if** self.art\_list[ind].get\_t\_num() == t\_num:

**del**(self.art\_list[ind])

**else**:

ind += 1

**def** del\_by\_name(self, name):

ind = 0

**while** ind < len(self.type\_list):

**if** self.type\_list[ind].get\_name() == name:

self.del\_articles\_by\_t\_num(self.type\_list[ind].get\_t\_num())

**del**(self.type\_list[ind])

**return** True

ind += 1

**return** False

**def** filter\_by\_price\_all(self):

res\_list = []

**for** item **in** self.type\_list:

t\_num = item.get\_t\_num()

is\_add = True

**for** article **in** self.art\_list:

**if** article.get\_t\_num() == t\_num **and** int(article.get\_price()) <= 10:

is\_add = False

**break**

**if** is\_add:

res\_list.append(item)

**return** res\_list

**def** save\_state(self):

*"""save structure of type's and articles' lists to pickle files"""*

self.pckl\_type = open(self.t\_file, **"wb"**)

self.pckl\_art = open(self.a\_file, **"wb"**)

pickle.dump(self.type\_list, self.pckl\_type)

self.pckl\_type.close()

pickle.dump(self.art\_list, self.pckl\_art)

self.pckl\_art.close()

**def** get\_id\_by\_name(self, type\_name):

*"""get type's identifier or return False if it's not found"""*

**for** item **in** self.type\_list:

**if** item.get\_name() == type\_name:

**return** item.get\_t\_num()

**return** False

**def** get\_t\_num\_articles(self, t\_num):

res = []

**for** item **in** self.art\_list:

**if** item.get\_t\_num() == t\_num:

res.append(item)

**return** res

**def** del\_article\_by\_name(self, art\_name):

ind = 0

**while** ind < len(self.art\_list):

**if** self.art\_list[ind].get\_name() == art\_name:

**del**(self.art\_list[ind])

**return** True

ind += 1

**return** False

**def** add\_articles\_lst(self, to\_add\_list):

self.art\_list += to\_add\_list

**def** edit\_article\_by\_name(self, to\_change, article):

*"""return True if article is edited, false return otherwise"""*

ind = 0

**while** ind < len(self.art\_list):

**if** self.art\_list[ind].get\_name() == to\_change:

self.art\_list[ind] = article

**return** True

ind += 1

**return** False

**def** edit\_type(self, name, \_type):

ind = 0

**while** ind < len(self.type\_list):

**if** self.type\_list[ind].get\_name() == name:

self.type\_list[ind] = \_type

**return** True

ind += 1

**return** False

**def** get\_next\_t\_num(self):

**return** self.type\_list[len(self.type\_list) - 1].get\_t\_num() + 1

**# controller.py**

**from model import \***

**import** view

database = Database(**"type.pcl"**, **"articles.pcl"**)

**def** form\_articles\_list(t\_num):

*"""form list of articles from input to add to the new type"""*

cond = **'y'**

art\_list = []

**while** cond == **'y'**:

art\_name, amount, price = view.fill\_article(**"article's"**)

**if** amount.isalpha() **or** price.isalpha():

view.invalid\_int\_input()

**continue**

article = Article(art\_name, amount, price, t\_num)

art\_list.append(article)

cond = view.enter\_cond()

**return** art\_list

**def** print\_all\_types():

view.print\_types(database.get\_type\_list())

**def** print\_all\_types\_with\_articles():

*"""Function is checked"""*

**for** item **in** database.get\_type\_list():

view.print\_type(item)

view.print\_articles(database.get\_t\_num\_articles(item.get\_t\_num()))

**def** print\_articles():

**for** article **in** database.get\_art\_list():

view.print\_article(article)

**def** find\_type\_by\_name():

to\_find = view.enter\_name(**"type"**)

view.print\_type(database.find\_type(to\_find))

**def** del\_type():

to\_find = view.enter\_name(**"type"**)

**if not** database.del\_by\_name(to\_find):

view.invalid\_input()

**def** add\_type():

name, country, vol = view.fill\_art\_type(**"type's"**)

t\_num = 0

t\_len = len(database.type\_list)

**if** t\_len > 0:

t\_num = database.get\_next\_t\_num()

art\_type = ArticleType(name, country, vol, t\_num)

database.get\_type\_list().append(art\_type)

articles\_list = form\_articles\_list(art\_type.get\_t\_num())

database.add\_articles\_lst(articles\_list)

**def** delete\_type():

view.type\_delete\_caution()

name = view.enter\_name(**"type"**)

**if not** database.del\_by\_name(name):

view.invalid\_type()

**def** exit\_case():

view.exit\_key()

database.save\_state()

**def** print\_greater\_10():

view.print\_types(database.filter\_by\_price\_all())

**def** edit\_article\_by\_name():

to\_change = view.enter\_name(**"article's"**)

art\_to\_change = database.find\_article(to\_change)

**if not** art\_to\_change:

view.invalid\_input()

**else**:

name, amount, price = view.fill\_article(**"new article's"**)

article = Article(name, amount, price, art\_to\_change.get\_t\_num())

database.edit\_article\_by\_name(to\_change, article)

**def** add\_article(type\_item):

to\_add\_list = form\_articles\_list(type\_item.get\_t\_num())

database.add\_articles\_lst(to\_add\_list)

**def** del\_article():

to\_del = view.enter\_name(**"article"**)

**if not** database.del\_article\_by\_name(to\_del):

view.invalid\_input()

**def** edit\_articles():

\_choice = view.edit\_articles()

**if** \_choice == **"1"**:

edit\_article\_by\_name()

**elif** \_choice == **"2"**:

**"""Add article bounded to the entered type"""**

name = view.enter\_name(**"type's"**)

type\_item = database.find\_type(name)

**if not** type\_item:

view.invalid\_input()

**else**:

add\_article(type\_item)

**elif** \_choice == **"3"**:

del\_article()

**else**:

view.invalid\_input()

**def** edit\_type():

to\_change = view.enter\_name(**"type's"**)

type\_to\_change = database.find\_type(to\_change)

**if not** type\_to\_change:

view.invalid\_input()

**else**:

name, country, vol = view.fill\_art\_type(**"new type's"**)

\_type = ArticleType(name, country, vol, type\_to\_change.get\_t\_num())

database.edit\_type(to\_change, \_type)

**def** main\_func():

choice = **''**

**while** choice != **"9"**:

choice = str(view.main\_menu())

**if** choice == **"1"**:

print\_all\_types\_with\_articles()

**elif** choice == **"2"**:

view.print\_types(database.type\_list)

**elif** choice == **"3"**:

print\_articles()

**elif** choice == **"4"**:

add\_type()

**elif** choice == **"5"**:

edit\_type()

**elif** choice == **"6"**:

delete\_type()

**elif** choice == **"7"**:

edit\_articles()

**elif** choice == **"8"**:

print\_greater\_10()

**elif** choice == **"9"**:

exit\_case()

**else**:

view.invalid\_input()

main\_func()

**view.py**

**def main\_menu():**

raw\_input(**"Press 'enter' to continue..."**)

**print "\n---Database type - articles ----"**

**print "--- Make a choice ---"**

**print "1 - Print all presented types with articles"**

**print "2 - Print types"**

**print "3 - Print articles"**

**print "4 - Add type"**

**print "5 - Edit type"**

**print "6 - Delete type"**

**print "7 - Edit articles"**

**print "8 - Print all articles with cost greater than 10"**

**print "9 - Exit the database"**

**return** raw\_input(**"\nEnter a choice:"**)

**def** print\_article(article):

**print "\tName:%s, amount:%s, price:%s, t\_num:%d "** \

% (article.get\_name(), article.get\_amount(), article.get\_price(), article.get\_t\_num())

**def** print\_articles(articles):

**for** article **in** articles:

print\_article(article)

**def** print\_type(item):

**if not** item:

**print "Sorry, no such item..."**

**else**:

**print "Type:%s, country:%s, volume:%s, t\_num:%d"** \

% (item.get\_name(), item.get\_country(), item.get\_vol(), item.get\_t\_num())

**def** print\_types(lst):

**for** item **in** lst:

print\_type(item)

**def** enter\_name(ident):

input\_val = **"Enter "** + ident + **" name:"**

**return** str(raw\_input(str(input\_val)))

**def** invalid\_input():

**print "Entered invalid value, try again"**

**def** invalid\_int\_input():

**print "Sorry, digit's value expected, try again"**

**def** invalid\_type():

**print "Sorry, no such type was found, try again"**

**def** exit\_key():

**return** raw\_input(**"Press enter to exit..."**)

**def** fill\_art\_type(prompt):

**return** (raw\_input(**"Enter %s name:"** % prompt),

raw\_input(**"Enter type's country:"**),

raw\_input(**"Enter type's overall volume:"**))

**def** fill\_article(prompt):

**return** (raw\_input(**"Enter %s name:"** % prompt),

raw\_input(**"Enter amount:"**),

raw\_input(**"Enter price:"**))

**def** fill\_list\_of\_a\_type():

**return** (raw\_input(**"\tEnter article's name:"**),

raw\_input(**"\tEnter amount:"**),

raw\_input(**"\tEnter price:"**),

raw\_input(**"\tContinue(y,n):"**))

**def** enter\_cond():

**return** raw\_input(**"\tContinue(y,n):"**)

**def** edit\_articles():

**print "\t\tEdit article by name - 1"**

**print "\t\tAdd new article - 2"**

**print "\t\tDelete article - 3"**

**return** raw\_input(**"Enter a choice:"**)

**def** type\_delete\_caution():

**print "Caution: all articles of the type will be gone"**

**---Database type - articles ----**

**--- Make a choice ---**

**1 - Print all presented types with articles**

**2 - Print types**

**3 - Print articles**

**4 - Add type**

**5 - Edit type**

**6 - Delete type**

**7 - Edit articles**

**8 - Print all types with minimal article's cost greater than 10**

**9 - Exit the database**

**Enter a choice:1**

**Type:Office suplies, country:China, volume:1500, t\_num:0**

**Name:pen, amount:120, price:5, t\_num:0**

**Name:copybook, amount:150, price:2, t\_num:0**

**Type:Fruit, country:Italy, volume:6000, t\_num:1**

**Name:banana, amount:500, price:8, t\_num:1**

**Name:orange, amount:700, price:12, t\_num:1**

**Name:peach, amount:40, price:21, t\_num:1**

**Type:Fabric, country:Turkey, volume:300, t\_num:3**

**Name:silk, amount:200, price:28, t\_num:3**

**Name:wool, amount:120, price:19, t\_num:3**

**Press 'enter' to continue…**

**Enter a choice:2**

**Type:Office suplies, country:China, volume:1500, t\_num:0**

**Type:Fruit, country:Italy, volume:6000, t\_num:1**

**Type:Vegetables, country:Turkey, volume:5200, t\_num:2**

**Type:Fabric, country:Turkey, volume:300, t\_num:3**

**Press 'enter' to continue…**

**Enter a choice:3**

**Name:pen, amount:120, price:5, t\_num:0**

**Name:copybook, amount:150, price:2, t\_num:0**

**Name:banana, amount:500, price:8, t\_num:1**

**Name:orange, amount:700, price:12, t\_num:1**

**Name:peach, amount:40, price:21, t\_num:1**

**Name:silk, amount:200, price:28, t\_num:3**

**Name:wool, amount:120, price:19, t\_num:3**

**Enter a choice:8**

**Type:Fabric, country:Turkey, volume:300, t\_num:3**

**Press 'enter' to continue...**