**Лабараторна №9**

**Варіант 2**

Завдання 1

**Склад**. База товарів, які зберігаються на складі: назва товару, одиниця виміру,

кількість. Організувати реєстрацію надходження товарів (формування накладної прийому)

і відвантаження (формування накладної видачі). Дані зберігаються в масиві.

Розв’язок:

class Item:

def \_\_init\_\_(self, name, unit, count):

self.name = name

self.unit = unit

self.count = count

def \_\_str\_\_(self):

return "{0} ({1} {2})".format(self.name, self.count, self.unit)

class Storage:

def \_\_init\_\_(self):

self.items = []

def deposit(self, item):

self.items.append(item)

print("[Storage] + Deposited {0}".format(item))

def withdraw(self, item\_name, count):

for i in range(len(self.items)):

it = self.items[i]

if it.name == item\_name:

it.count -= count

if it.count < 0:

del self.items[i]

break

print("[Storage] - Withdrawed {0} by {1}".format(item\_name, count))

storage = Storage()

storage.deposit(Item("Potato", "kg", 50))

storage.withdraw("Potato", 25)

Завдання 2

**Довідник кухарки**. База даних продуктів: назва продукту, одиниці виміру, наявна

кількість продукту, вартість одиниці. Організувати вибір за довільним запитом. Дані

зберігаються в масиві записів, який створюється динамічно.

Розв’язок:

class Product:

def \_\_init\_\_(self, name, unit, count, cost):

self.name = name

self.unit = unit

self.count = count

self.cost = cost

def \_\_str\_\_(self):

return "{0} ({1} {2}, {3} per {2})".format(self.name, self.count, self.unit, self.cost)

class Storage:

def \_\_init\_\_(self):

self.products = []

def add\_product(self, p):

self.products.append(p)

def get\_all\_products(self):

return self.products

def searcher(self, item, criteria, query):

if criteria == "name":

return query in item.name

elif criteria == "count":

return float(query) <= item.count

elif criteria == "cost":

return abs(float(query) - item.cost) < 0.0001

else:

return False

def search(self, criteria, query):

return [item for item in self.products if self.searcher(item, criteria, query)]

############################

storage = Storage()

n = int(input("How many products? "))

for i in range(n):

name = input("Product #{0}, Name = ".format(i+1))

unit = input("Product #{0}, Unit = ".format(i+1))

count = float(input("Product #{0}, Count = ".format(i+1)))

cost = float(input("Product #{0}, Cost = ".format(i+1)))

storage.add\_product(

Product(name, unit, count, cost)

)

print("\n")

print("[Storage]")

for p in storage.get\_all\_products():

print(p)

print("\n[Search]")

criteria = input("Search criteria (name, count, cost): ")

query = input("Search query: ")

print("RESULTS:")

for p in storage.search(criteria, query):

print(p)

Приклад:

**How many products? 3**

**Product #1, Name = Potato**

**Product #1, Unit = kg**

**Product #1, Count = 50**

**Product #1, Cost = 4**

**Product #2, Name = Milk**

**Product #2, Unit = l**

**Product #2, Count = 30**

**Product #2, Cost = 7**

**Product #3, Name = Carrot**

**Product #3, Unit = kg**

**Product #3, Count = 15**

**Product #3, Cost = 6**

**[Storage]**

**Potato (50.0 kg, 4.0 per kg)**

**Milk (30.0 l, 7.0 per l)**

**Carrot (15.0 kg, 6.0 per kg)**

**[Search]**

**Search criteria (name, count, cost): name**

**Search query: Carrot**

**RESULTS:**

**Carrot (15.0 kg, 6.0 per kg)**