Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Програмування

Лабораторна робота №5

«Основи об’єктно-орієнтованого програмування. Модулі та пакети»

Виконав:

студент групи ІО-64

Андрійчук Д. А.

Залікова книжка №ІО 6401

Перевірив Новотарський М. А.

Київ 2016

**Тема:** «Основи об’єктно-орієнтованого програмування. Модулі та пакети».

**Мета:** вивчити способи створення та підключення модулів та пакетів. Основи ООП. Методи і атрибути класів та робота з ними. Побудова програми у стилі ООП.

**Завдання:** 1. Вивчити матеріал лекцій 18, 19, 20 та 21.

2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

**Короткі теоретичні відомості**:

**Модуль datetime** предоставляет классы для обработки времени и даты разными способами.

**datetime.now**(tz=None) - объект datetime из текущей даты и времени системы.

**Модуль calendar** позволяет напечатать себе календарик (а также содержит некоторые другие полезные функции для работы с календарями).

**calendar.weekday**(year, month, day) - день недели для этой даты.

**Елементи ООП.**

Клас це об'єкт, типом якого є тип type:

**class** Name [(<Клас1>[, ... , , ... , <Класn>])]:

<атрибути>

<методи>

Об'єкт - це будь-яка сутність в Python

Створення метода класу. Метод всередині класу створюється так само, як і звичайна функція, за допомогою інструкції def

**class** Name:

def method(self,[]):

<команди>

Інкапсуляція – властивість мови програмування, яка дозволяє користувачеві не задумуватися про складність реалізації програмного забезпечення, яке використовується (не думати про внутрішні принципи його роботи), а взаємодіяти з ним шляхом використання інтерфейсу

Роздруківка фрагменту тексту програми.

Відповідно до номера у списку вибрати індивідуальне завдання. Написати програму. Забезпечити ввід даних з клавіатури комп’ютера та друк результатів.

Створити клас, який описує роботу кафе і містить асортимент страв, їх базову ціну, знижки за днями тижня, знижки для постійних відвідувачів та метод обчислення вартості замовлення. Створити об’єкт даного класу, що дозволяє визначити вартість обіду для довільного клієнта з урахуванням знижок. Програма повинна виводити рахунок зі списком замовлених страв, їх ціною та знижкою з урахуванням особи клієнта та дня тижня

**import** calendar, datetime  
**class** Price:  
 menu = {**"Fire Grilled Cobb Salad"**: 10.99, **"Chicken Caesar Salad"**: 10.29, **"Roast chicken"**: 25.25,  
 **"Broiled Sirloin Tips"**: 14.99, **"Mushroom Bleu Top Sirloin"**: 11.79, **"Baked Schrod"**: 12.49,  
 **"Fresh Glazed Salmon"**: 12.99}  
 sale = {0: 1, 1: 0, 2: 10, 3: 25, 4: 5, 5: 2, 6: 1}*# 0 - "Monday", 1 - "Tuesday", ..., 6 - "Sunday"* clients = [**"Андрійчук"**, **"Бандурін"**, **"Бровченко"**, **"Кожевніков"**, **"Фащевський"**, **"Фельдман"**, **"Ханін"**]  
 dishes\_cost = []  
 total = 0  
 **def** bill(self, order, day, person=**None**):  
 **for** i **in** range(len(order)):  
 self.dishes\_cost.append(self.menu[order[i]])  
  
 **for** i **in** range(len(order)):  
 self.dishes\_cost[i] -= self.dishes\_cost[i]\*self.sale[day] / 100  
  
 **if** person **in** self.clients:  
 self.dishes\_cost[i] \*= 0.9  
 **for** i **in** range(len(order)):  
 print(**"%s : %.2f"** % (order[i], self.dishes\_cost[i]))  
 self.total += self.dishes\_cost[i]  
 print(**"Total : %.2f"** % self.total)  
first = Price()  
order = []  
new\_dish = **""**number\_of\_dish = 1  
**for** key **in** sorted(first.menu.keys()):  
 print(**"%s. %s - %s"** % (number\_of\_dish, key, first.menu[key]))  
 number\_of\_dish += 1  
print(**"Введіть stop для зупинки вводу"**)  
**while** new\_dish.upper() != **'STOP'**:  
 new\_dish = (input(**'Name of dish: '**)).lower().title()  
 **if** new\_dish.upper() == **'STOP'**:  
 **pass  
 else**:  
 **if** new\_dish **in** Price.menu:  
 order.append(new\_dish)  
 **else**:  
 print(**"ERROR"**)  
now = datetime.datetime.now()  
person = input(**'Last name of client: '**).lower().title()

print(**"\n"**)  
first.bill(order, calendar.weekday(now.year, now.month, now.day), person)

Результат роботи

1. Baked Schrod - 12.49

2. Broiled Sirloin Tips - 14.99

3. Chicken Caesar Salad - 10.29

4. Fire Grilled Cobb Salad - 10.99

5. Fresh Glazed Salmon - 12.99

6. Mushroom Bleu Top Sirloin - 11.79

7. Roast chicken - 25.25

Введіть stop для зупинки вводу

Name of dish: bROILED sirloin tips

Name of dish: Baked Schrod

Name of dish: er

ERROR

Name of dish: Fire Grilled Cobb Salad

Name of dish: stop

Last name of client: андрійчук

Broiled Sirloin Tips : 13.36

Baked Schrod : 11.13

Fire Grilled Cobb Salad : 9.79

Total : 34.28

Висновок:

Одною з переваг ООП є краща модульність програмного забезпечення (тисячу функцій процедурної мови, в ООП можна замінити кількома десятками класів із своїми методами). На відміну від традиційних поглядів, коли програму розглядали як набір підпрограм, або як перелік інструкцій комп'ютеру, ООП програми можна вважати сукупністю об'єктів. Відповідно до парадигми об'єктно-орієнтованого програмування, кожний об'єкт здатний отримувати повідомлення, обробляти дані, та надсилати повідомлення іншим об'єктам. Кожен об'єкт — своєрідний незалежний автомат з окремим призначенням та відповідальністю.