



الجمهورية العربية السورية

وزارة التعليم العالي والبحث العلمي

جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة

## Second Network Programming Homework

إعداد الطلاب:

مرهف جابر عيده 2848

صالح مروان شهيلة 2910

المهدي يوسف مصطفى 2907

إشراف الدكتور المهندس:

مهند عيسى

## Question 1: Bank ATM Application with TCP Server/Client and Multi-threading

برنامج السيرفر:

```
import socket, threading

HOST = '0.0.0.0'
PORT = 3434

accounts = {
    "morhaf": 10050,
    "saleh": 7000,
    "almahde": 2000
}

def handle_client(conn, addr):
    print("Connected by {}".format(addr))
    while True:
        data = conn.recv(1024).decode()
        if not data:
            break

        account_number, operation, amount = data.split()
        if account_number not in accounts:
            conn.sendall("Invalid account Name".encode())
            continue

        try:
            amount = float(amount)
        except ValueError:
            conn.sendall("Invalid amount".encode())
            continue

        if operation == "check_balance":
            balance = accounts[account_number]
            conn.sendall("Your balance is: {}".format(balance).encode())
        elif operation == "deposit":
            accounts[account_number] += amount
            conn.sendall("Deposit successful. New balance: {}".format(accounts[account_number]).encode())
        elif operation == "withdraw":
            if accounts[account_number] < amount:
                conn.sendall("Insufficient funds".encode())
            else:
```

```

        accounts[account_number] -= amount
        conn.sendall("Withdrawal successful. New balance:
{}".format(accounts[account_number]).encode())
    else:
        conn.sendall("Invalid operation".encode())

    conn.close()
    print("Client {} disconnected".format(addr))

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    print("Server listening on {}:{}".format(HOST, PORT))
    while True:
        conn, addr = s.accept()
        thread = threading.Thread(target=handle_client, args=(conn, addr))
        thread.start()

```

برامج العملاء:

```

import socket

HOST = '127.0.0.1'
PORT = 3434

while True:
    account_number = input("Enter your account Name: ")
    operation = input("Enter operation (check_balance, deposit, withdraw): ")
    if operation in ("deposit", "withdraw"):
        amount = float(input("Enter amount: "))
        data = f"{account_number} {operation} {amount}".encode()

        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.connect((HOST, PORT))
            s.sendall(data)
            response = s.recv(1024).decode()
            print(response)

```

هذا الكود هو خادم بسيط يستخدم مكتبة socket لتوفير اتصال TCP. يهدف الكود إلى تنفيذ نظام بسيط لإدارة حسابات المستخدمين، حيث يمكن للعملاء (clients) الاتصال بالخادم وإجراء عمليات مثل فحص الرصيد، وإيداع أو سحب الأموال. الكود يستخدم threading لتعدد المهام، حيث يتم إنشاء مؤشر لكل عميل يتصل بالخادم.

شرح أكواد الزبائن:

يستخدم مكتبة socket لإنشاء اتصال TCP بين العميل والخادم، ويستخدم threading لتعدد المهام. يسمح للمستخدمين بإدخال رقم الحساب والعمليّة المطلوبة مثل فحص الرصيد، وإيداع أو سحب الأموال.

```

import socket

HOST = '127.0.0.2'
PORT = 3434

while True:
    account_number = input("Enter your account Name: ")
    operation = input("Enter operation (check_balance, deposit, withdraw): ")
    if operation in ("deposit", "withdraw"):
        amount = float(input("Enter amount: "))
        data = f"{account_number} {operation} {amount}".encode()

        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.connect((HOST, PORT))
            s.sendall(data)
            response = s.recv(1024).decode()
            print(response)

```

```

import socket

HOST = '127.0.0.3'
PORT = 3434

while True:
    account_number = input("Enter your account Name: ")
    operation = input("Enter operation (check_balance, deposit, withdraw): ")
    if operation in ("deposit", "withdraw"):
        amount = float(input("Enter amount: "))
        data = f"{account_number} {operation} {amount}".encode()

        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.connect((HOST, PORT))
            s.sendall(data)
            response = s.recv(1024).decode()
            print(response)

```

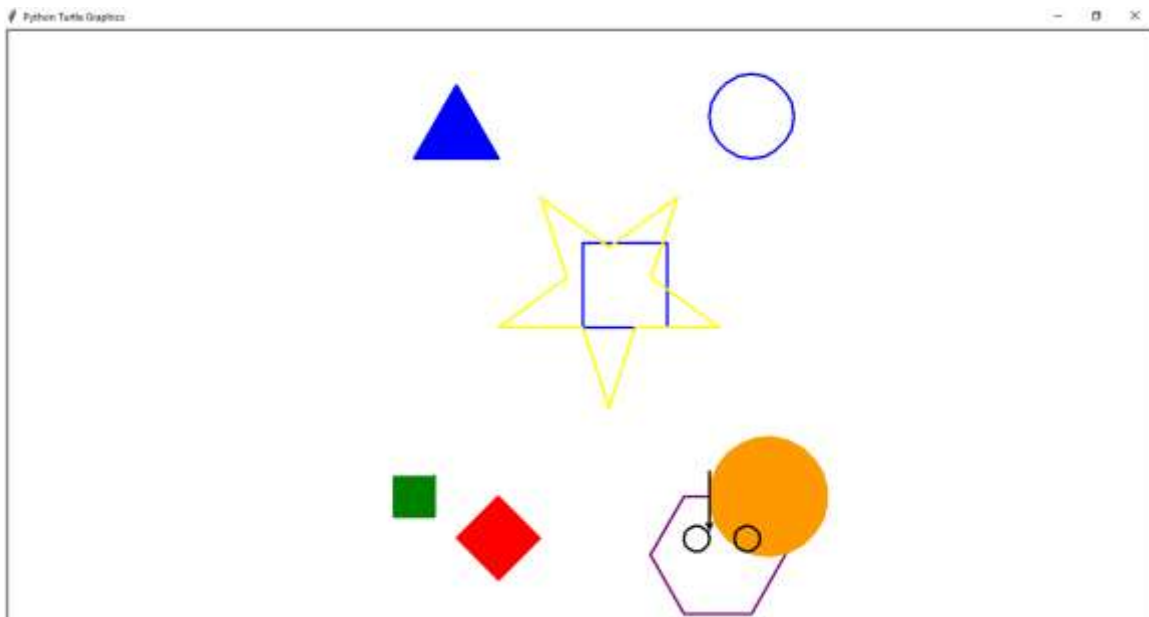
نتيجة التنفيذ: في السيرفر:

Server listening on 0.0.0.0:3434  
Connected by(1511 ,'127.0.0.1')  
Client ('127.0.0.1', 1511) disconnected  
Connected by(1512 ,'127.0.0.1')  
Client ('127.0.0.1', 1512) disconnected  
Connected by(1513 ,'127.0.0.1')  
Client ('127.0.0.1', 1513) disconnected

نتيجة التنفيذ: في العميل:

Enter your account Name: morhaf  
Enter operation (check\_balance, deposit, withdraw): withdraw  
Enter amount: 500  
Withdrawal successful. New balance: 9550.0  
Enter your account Name: saleh  
Enter operation (check\_balance, deposit, withdraw): check\_balance  
Your balance is: 7000  
Enter your account Name: almahde  
Enter operation (check\_balance, deposit, withdraw): deposit  
Enter amount: 800  
Deposit successful. New balance: 2800.0  
Enter your account Name:

## Question 2: Drawings program using python module turtle:



```
import turtle

t = turtle.Turtle()
t.speed(1)
t.color("blue")
t.pensize(3)

for i in range(4):
    t.forward(100)
    t.left(90)

t.penup()
t.goto(200, 200)
t.pendown()

t.circle(50)

t.penup()
t.goto(-200, 200)
t.pendown()

t.begin_fill()
for i in range(3):
    t.forward(100)
    t.left(120)
```

```
t.end_fill()

t.penup()
t.goto(-200, -200)
t.pendown()

t.color("green")
for i in range(50):
    t.forward(i)
    t.left(90)

t.penup()
t.goto(0, 0)
t.pendown()
t.color("yellow")
for i in range(5):
    t.forward(100)
    t.right(144)
    t.forward(100)
    t.left(72)

t.penup()
t.goto(200, -200)
t.pendown()
t.color("purple")
for i in range(6):
    t.forward(80)
    t.left(60)

t.penup()
t.goto(-100, -200)
t.pendown()
t.color("red")
t.begin_fill()
t.left(45)
t.forward(70)
t.left(90)
t.forward(70)
t.left(90)
t.forward(70)
t.left(90)
t.forward(70)
t.left(135)
t.end_fill()
```



```
t.penup()
t.goto(150, -200)
t.pendown()
t.color("#FF9900") # Orange
t.begin_fill()
t.circle(70)
t.end_fill()
t.penup()
t.goto(120, -250)
t.pendown()
t.color("black")
t.circle(15)
t.penup()
t.goto(180, -250)
t.pendown()
t.circle(15)
t.penup()
t.goto(150, -170)
t.pendown()
t.forward(70)

turtle.done()
```