

Rešenja prvog kolokvijuma iz Operativnih sistema 2 Decembar 2019.

1. (10 poena)

Trenutak [ms]	Izvršava se proces
15	C
25	B
35	D
55	D
65	C
75	D
95	B
110	D
125	A

2. (10 poena)

```
monitor DiningPhilosophers;
export startEating, stopEating;

var
    forks : array 0..4 of boolean;
    canEat : array 0..4 of condition;

procedure startEating (i : integer);
begin
    var left, right : integer;
    left := i; right := (i+1) mod 5;
    while forks[left] or forks[right] do canEat[i].wait;
    forks[left] := true;
    forks[right] := true;
end;

procedure stopEating (i : integer);
begin
    var left, right : integer;
    left := i; right := (i+1) mod 5;
    forks[left] := false;
    forks[right] := false;
    canEat[left].signal;
    canEat[right].signal;
end;

begin
    var i: integer;
    for i:=0 to 4 do forks[i] := false;
end;
```

3. (10 poena)

```
public class Server {
    private final static int N = 100;
    private final ServerSocket socket;
    private final ContentPrice contentPrice;
    private final Map<String, Integer> accounts = new HashMap<String,
Integer>();

    public Server() throws IOException {
        socket = new ServerSocket(5555);
        contentPrice = new ContentPrice();
    }

    public void work() throws IOException {
        while(true) {
            Socket client = socket.accept();

            new RequestHandler(client, this, contentPrice).start();
        }
    }

    public synchronized boolean withdraw(String user, int price) {
        if (!accounts.containsKey(user)) {
            accounts.put(user, N);
        }

        int amount = accounts.get(user);
        if (amount < price) {
            return false;
        }

        accounts.put(user, amount - price);
        return true;
    }
}

public class RequestHandler extends Thread {
    private final Socket client;
    private final Server server;
    private final ContentPrice contentPrice;

    public RequestHandler(Socket client, Server server, ContentPrice
contentPrice) {
        this.client = client;
        this.server = server;
        this.contentPrice = contentPrice;
    }

    public void run() {
        Service service = new Service(client);
        String user = service.receiveMessage();
        while(true) {
            String content = service.receiveMessage();

            if (content.equals("End")) {
                break;
            }

            if (server.withdraw(user, contentPrice.getPrice(content))) {
                // Deliver content to user
            }
        }
    }
}
```

```
        } else {
            service.sendMessage("Insufficient funds");
            break;
        }
    }
}
}
```

Klasa Service (Usluga) je data na vežbama.