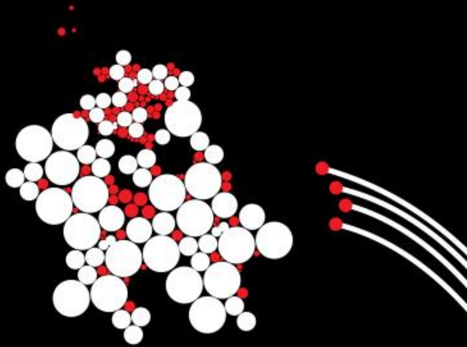
UNIVERSITY OF TWENTE.



Threads in Java: an example

Topic of Software Systems (TCS module 2)

Lecturer: Marieke Huisman

EXAMPLE: BANK ACCOUNT

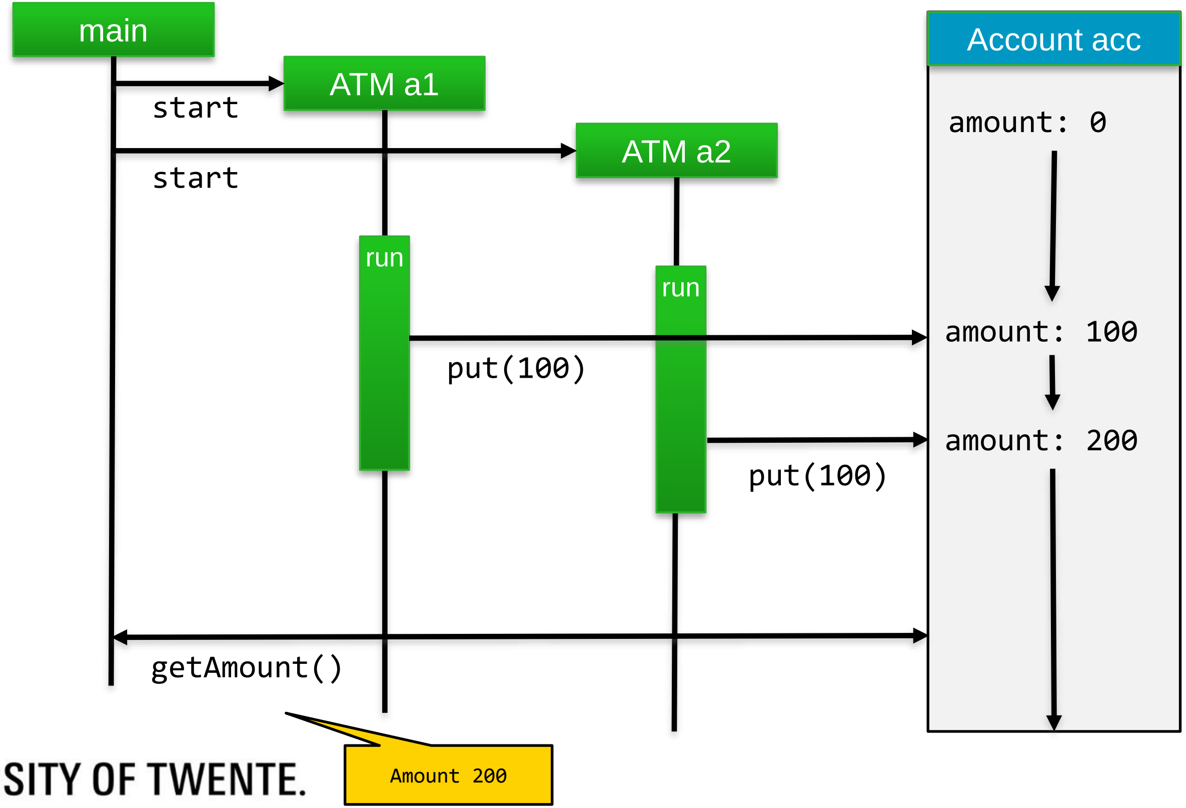
```
public class BankAccount {
    private double amount = 0.0;
    public void deposit(double val) { amount = amount + val; }
    public void withdraw(double val) { amount = amount - val; }
    public double getAmount() { return amount; }
}
```

WHAT WILL THIS PROGRAM PRINT?

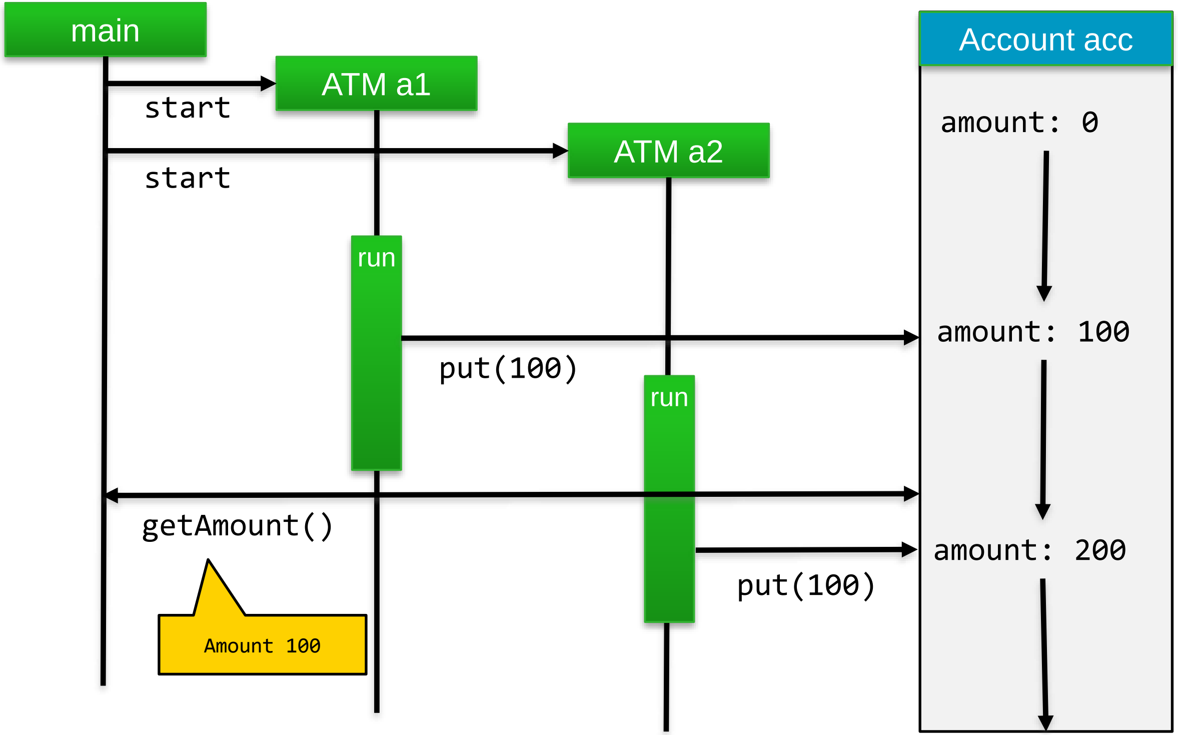
```java
public class Deposit100ATM implements Runnable {
    private BankAccount acc;
    public Deposit100ATM(BankAccount acc) { this.acc = acc; }
    public void run() { acc.deposit(100.0); }
}

public static void main(String[] args) {
    BankAccount acc = new BankAccount();
    Deposit100ATM a1 = new Deposit100ATM(acc);
    Deposit100ATM a2 = new Deposit100ATM(acc);
    new Thread(a1).start();
    new Thread(a2).start();
    System.out.println("Amount on account is " + acc.getAmount());
}
```

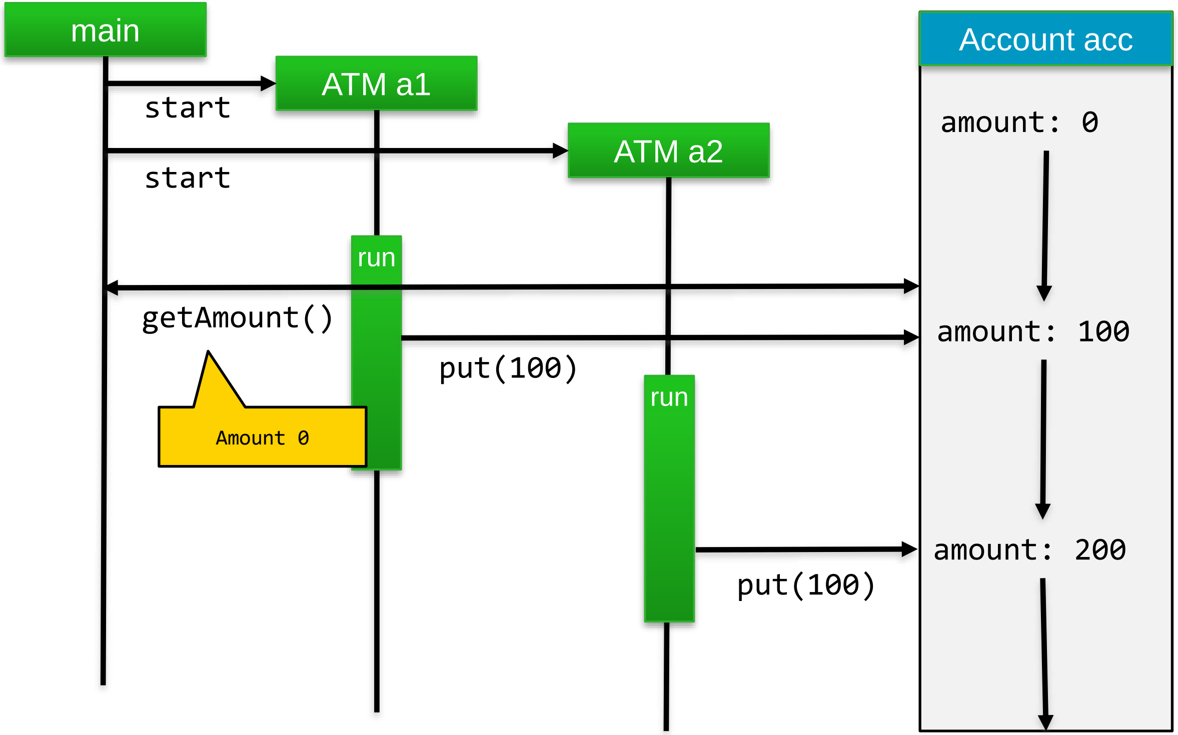
A POSSIBLE EXECUTION



A POSSIBLE EXECUTION



A POSSIBLE EXECUTION



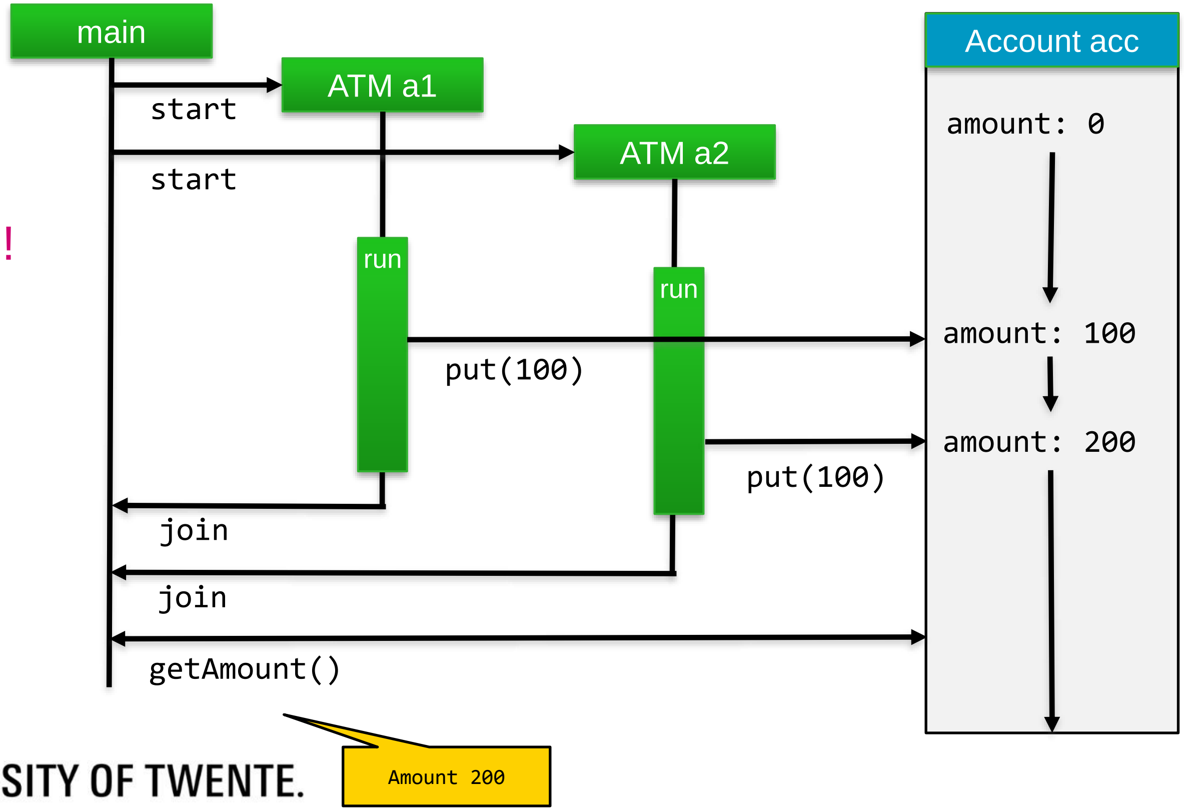
UNIVERSITY OF TWENTE.

WHAT WILL THIS PROGRAM PRINT?

```java
public static void main(String[] args) {
    BankAccount acc = new BankAccount();
    Deposit100ATM a1 = new Deposit100ATM(acc);
    Deposit100ATM a2 = new Deposit100ATM(acc);
    a1.start();
    a2.start();
    try {
        a1.join();
        a2.join();
    } catch (InterruptedException e) { ... }
    System.out.println(”Amount on account is " + acc.getAmount());
}
```

A POSSIBLE EXECUTION

Always?
No, not always!


main
ATM a1
ATM a2
Account acc
start
start
amount: 0
run
run
put(100)
amount: 100
amount: 200
put(100)
join
join
getAmount()
Amount 200