

```

import database.DataBase;
import database.DbItem;
import transactions.DeadlockTransaction;
import transactions.NoDeadlockTransaction;
import transactions.Transaction;

import java.util.concurrent.TimeUnit;

public class SGBD {
    public static void main(String[] args) throws InterruptedException {
        //executeSerialSchedule();
        //executeConcurrentScheduleWithNoDeadLockPrevention();
        executeConcurrentScheduleWithDeadLockPrevention();
    }

    public static void executeSerialSchedule(){
        System.out.println("Executing serial schedule...");

        DataBase dataBase = new DataBase("Alunos");
        DbItem itemA = dataBase.getItemFromDataBase("Nome", 1, "Bianca");
        DbItem itemB = dataBase.getItemFromDataBase("Nome", 2, "Lucas");

        Transaction transaction1 = new DeadlockTransaction(itemA, itemB);
        Transaction transaction2 = new DeadlockTransaction(itemB, itemA, transaction1);

        transaction1.start();
        transaction2.start();
    }

    public static void executeConcurrentScheduleWithNoDeadLockPrevention(){
        System.out.println("Executing concurrent schedule with no deadlock prevention...");

        DataBase dataBase = new DataBase("Alunos");
        DbItem itemA = dataBase.getItemFromDataBase("Nome", 1, "Bianca");
        DbItem itemB = dataBase.getItemFromDataBase("Nome", 2, "Lucas");

        Transaction transaction1 = new DeadlockTransaction(itemA, itemB);
        Transaction transaction2 = new DeadlockTransaction(itemB, itemA);

        transaction1.start();
        transaction2.start();
    }

    public static void executeConcurrentScheduleWithDeadLockPrevention() throws InterruptedException {
        System.out.println("Executing concurrent schedule with deadlock prevention...");

        DataBase dataBase = new DataBase("Alunos");
        DbItem itemA = dataBase.getItemFromDataBase("Nome", 1, "Bianca");
        DbItem itemB = dataBase.getItemFromDataBase("Nome", 2, "Lucas");

        Transaction transaction1 = new NoDeadlockTransaction(itemA, itemB);
        Transaction transaction2 = new NoDeadlockTransaction(itemB, itemA);

        transaction2.start();

        TimeUnit.MILLISECONDS.sleep(30);

        transaction1.start();
    }
}

```

