Tentamen
Nätverksprogrammering
Del 2

2018–04–11, 8.00–13.00

Tillåtna hjälpmedel för denna del av tentamen:

- Java snabbreferens.
- Kursboken: Java Network Programming av Eliotte Rusty Harold.
- Valfri lärobok i Java.
- Utskrift av OH-bilder från föreläsningarna.

Denna tentamen i kursen Nätverksprogrammering består av två delar – en del som innehåller frågor av teoretisk/principiell/utredande karaktär och en del som innehåller praktiska programmeringsuppgifter. Detta är del 2. Den ska du ha erhållit tillsammans med ett färgat tentamensomslag när du lämnade in din lösning på del 1 av tentamen.

För godkänt betyg på tentamen krävs sammanlagt minst 20 poäng på tentamen, varav minst 8 poäng på vardera deltentamen.

---

1. **Budgivning**

Auktioner är ett populärt sätt att köpa och sälja begagnade prylar. Din uppgift är att implementera delar av budgivningen för en online-auktion. I klienten visas alla objekt som är till salu, en lista över senaste buden och det finns möjlighet att lägga bud. Delarna som interagerar med användaren ska du inte implementera. Din uppgift är istället att implementera kommunikationen för budgivningen och större delen av serven. På klientsidan ligger den koden i klassen `AuctionCommunication`. Bud representeras av klassen `Bid`, som är tillgänglig både på klienten och serven.

```
public class Bid {

    // your own attributes and methods, if needed

    public int objectId;
    public int amount;

    Bid(int objectId, int amount){
        this.objectId = objectId;
        this.amount = amount;
    }

    int getObjectId(){
        return objectId;
    }

    int getAmount(){
        return amount;
    }
}
```

```
public class AuctionCommunication {
    /**
     *    FOR YOU TO IMPEMENT
     **/

    // your own attributes and methods

    public AuctionCommunication() {
        // any setup needed
        // YOUR CODE
    }

    public boolean sendNewBid(Bid bid) {
        // Send a bid to the server. Returns true if the bid was accepted,
        // i.e. higher than earlier bids
        // YOUR CODE
    }

    public Bid awaitNewBid() {
        // blocks until a new bid is available
        // YOUR CODE
    }
}
```

Klienten använder `AuctionCommunication` för att kommunicera med auktionsfirmans server. När köparen lägger ett bud anropar programmet `AuctionCommunication.sendNewBid(Bid bid)`. Metoden skickar budet till serven och får som svar ett booleskt värde som anger om budet accepterades (någon annan kan ju ha lagt ett högre bud utan att köparen sett det).

Programmet ska presentera alla bud som accepteras (inte bara egna bud, utan alla bud på alla objekt, av alla köpare). Buden hämtas från serven via `AuctionCommunication.awaitNewBid()`. Metoden returnerar ett bud i taget. Metoden ska blockera om det inte finns nya bud.

Serven ska alltid kunna ta emot nya bud (tråden som tar emot bud får inte blockeras under längre tider vilket t.ex. kan ske om den skickar data över nätverk). När ett nytt bud kommer in ska serven acceptera eller avböja det. Logiken för det finns i klassen `Auctioneer`. *Den klassen behöver du inte implementera.* Accepterade bud ska även skickas till alla klienter, inklusive den som la budet. Det kan hända att en klient under korta perioder inte hämtar bud i samma takt som nya kommer in. Det är ditt ansvar att säkerställa att alla klienter får alla accepterade bud, inte bara det senaste. Du kan anta att detta bara sker under kortare perioder (antalet bud som inte hämtats är mindre än ca 100).

```
class Auctioneer{
    /**
      * DO NOT IMPLEMENT
      * A helper class that decides if a bid is accepted.
      * Must run on the server and all bids must be registered
      * (do not use multiple instances of this class)
     **/
    public boolean makeBid(Bid bid) { /* code you do not need to implement */ }
}
```

Designa och beskriv protokollet mellan serven och klienten (när sker uppkoppling, vilken data skickas och i vilken ordning). Implementera klassen `AuctionCommunication` samt serverprogramet. All kod på serversidan, förutom `Auctioneer`, ska implementeras. På klientsidan ska du bara implementera `AuctionCommunication` och eventuella trådar, klasser och/eller hjälpmetoder som den använder.

(20p)

Slut!