

Name:

Punkte:

Note:

Klausur zum Modul „Fortgeschrittene Programmiertechniken“, SS 2019 (8 Aufgaben auf 10 Seiten, maximal 60 Punkte)

- 1. Aufgabe (2[a]+2[b]=4 Punkte): Nutzung eines Objekts durch mehrere Threads:
Gegeben sei folgende Klasse:

```
public class Multiplication
{
    private int result;

    public int multiply(int[] args)
    {
        result = 1;
        for(int arg: args)
        {
            result *= arg;
        }
        return result;
    }
}
```

a) Welche Probleme gibt es, wenn ein Objekt dieser Klasse parallel von mehreren Threads genutzt wird? Geben Sie **ein ganz konkretes Beispiel** zur Erläuterung des Problems! Ignorieren Sie dabei ein mögliches Überlaufen von result; dieses Problem ist nicht gemeint!

b) Wie können Sie das Problem durch Änderung des Codes **ohne Synchronisation** lösen, auch wenn alle Threads dasselbe Multiplication-Objekt benutzen?

- 2. Aufgabe (6[a]+8[b]+2[c]=16 Punkte): wait / notify / notifyAll:

In dieser Aufgabe geht es um eine logische Uhr, welche durch die Klasse LogicalTime realisiert werden soll. Die Klasse hat zwei Methoden:

- tick (Rückgabetyp void, ohne Parameter): damit wird signalisiert, dass die Uhr einmal ticken soll
- waitTicks (Rückgabetyp void, ein int-Parameter): damit wartet der aufrufende Thread auf so viele Ticks, wie der Parameter angibt (d.h. wenn ein Thread t diese Methode mit dem Parameter n aufruft, dann muss von einem oder mehreren anderen Threads die Methode tick n Mal aufgerufen werden, damit der Thread t wieder aufwacht und aus der Methode waitTicks zurückkehrt – die logische Uhr tickt also nicht von selbst, sondern nur dann, wenn man tick aufruft).

Natürlich sollen beide Methoden von beliebig vielen Threads gleichzeitig ausgeführt werden können.

a) Was halten Sie von dieser Lösung? Funktioniert sie wie vorgegeben? Begründen Sie Ihre Antwort!

```
public class LogicalTime
{
    public synchronized void tick()
    {
        notifyAll();
    }

    public synchronized void waitTicks(int waitingTicks)
    {
        int ticks = waitingTicks;
        while(ticks > 0)
        {
            try
            {
                wait();
                ticks--;
            }
            catch(InterruptedException e)
            {
            }
        }
    }
}
```

b) Geben Sie eine bessere Lösung als in Teilaufgabe a an! Ignorieren Sie einen möglichen Überlauf von Zählvariablen!

```
public class LogicalTime
{
->
->
->
->
    public synchronized void tick()
    {
->
->
->
->
->
->
->
->
->
    }
    public synchronized void waitTicks(int waitingTicks)
    {
->
->
->
->
->
->
->
->
->
->
->
->
->
    }
}
```

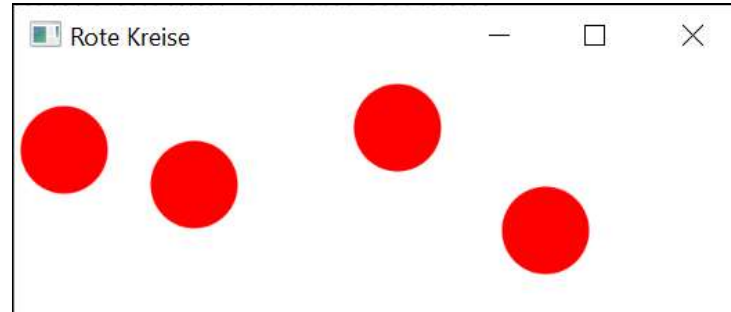
c) Brauchen Sie unbedingt notifyAll oder reicht notify? Bitte begründen Sie Ihre Antwort!

- 4

- ```
CheckBox cb = new CheckBox("Ich mag FOPT!!!!");
```

- 6. Aufgabe (7 Punkte): Grafikprogrammierung mit JavaFX:

In dieser Aufgabe geht es um eine Anwendung, mit der rote Kreise gezeichnet werden können:



Wenn man in das Fenster klickt, erscheint ein roter, *nicht ausgefüllter* Kreis mit  $RADIUS = 20$ . Der Mittelpunkt des Kreises befindet sich an der Position des Mauszeigers. Wenn man die Maus mit gedrückter Maustaste bewegt, bewegt sich der Mittelpunkt der Kreises entsprechend mit (Methoden `setCenterX` und `setCenterY`), der Radius ändert sich dabei nicht. Lässt man die Maustaste los, ist der Kreis ausgefüllt und kann nicht mehr verändert werden. Vervollständigen Sie den unten stehenden Programmcode auf der folgenden Seite!

```
package ... und import ... //nicht zu ergänzen
public class DrawingCircles extends Application
{
 private static final double RADIUS = 20;
 private Pane graphicsPane;
 private Circle c;

 public void start(Stage primaryStage)
 {
 graphicsPane = new Pane();
 graphicsPane.setOnMousePressed(
 (e -> mousePressed(e.getX(), e.getY()))
);
 graphicsPane.setOnMouseDragged(
 (e -> mouseDragged(e.getX(), e.getY()))
);
 graphicsPane.setOnMouseReleased(
 (e -> mouseReleased())
);
 primaryStage.setTitle("Rote Kreise");
 primaryStage.setScene(new Scene(graphicsPane, 350, 120));
 primaryStage.show();
 }
 public static void main(String[] args)
 {
 launch(args);
 }
}
```



- 7. Aufgabe (6[a]+3[b1]+2[b2]+5[b3]=16 Punkte): Sockets – TCP:
  - a) Geben Sie Programmcode an, in dem auf dem Port 7777 auf einen Verbindungsaufbau gewartet wird! Anschließend sollen beliebig viele Zeilen von ASCII-Daten gelesen werden (jeweils mit einem Newline-Symbol abgeschlossen), bis die Verbindung von der anderen Seite geschlossen wird. Es sollen keine Daten gesendet werden. Als Rahmen um den Code muss weder eine Klasse noch eine Methode angegeben werden. Es soll nicht die Klasse TCPSocket aus der Vorlesung, sondern es sollen nur Klassen aus der Java-Klassenbibliothek verwendet werden.



b) Angenommen, in einer Client-Server-Anwendung über TCP wird das typische Request-Response-Muster genutzt (wie in unserem Zählerbeispiel aus der Vorlesung, nicht wie im Beispiel aus Teilaufgabe a). Das heißt: Während des Bestehens einer TCP-Verbindung sendet der Client eine Anfrage (Request), die vom Server beantwortet wird (Response).

b1) Was ist in der Regel schneller bei der Nutzung eines sequenziellen Servers: Wenn der Client immer auf eine Antwort wartet, bevor er den nächsten Request sendet (send – receive – send – receive – send – receive ...) oder wenn der Client alle Requests sendet und dann auf eine Antwort wartet (send – send – send – ... receive – receive – receive ...)? Oder spielt das keine Rolle? Begründen Sie Ihre Antwort! Gehen Sie hier erst einmal davon aus, dass es zu keinen Problemen kommt, sondern dass alles funktioniert! Wie ist es dann mit der Geschwindigkeit?

b2) Wie ist die Antwort auf die Frage aus Teilaufgabe b1, wenn ein statisch paralleler oder dynamisch paralleler Server wie in der Vorlesung verwendet wird?

b3) Bei der Nutzung des Burst-Modus (send – send – send – ... receive – receive – receive ...) kann es zu einem Problem kommen, wenn sehr viele Nachrichten gesendet werden, bevor man anfängt die Antworten aufzusammeln. Beschreiben Sie das Problem (Ursache und Wirkung)? Wie kann man das Problem vermeiden?

- 8. Aufgabe (2[a]+2[b]=4 Punkte: RMI – Stub und Skeleton:
  - a) Geben Sie an, welchen Code ein RMI-Stub enthält! Also: Welche Methoden besitzt eine Stub-Klasse und welche Funktion haben diese Methoden im Wesentlichen?
  
  
  
  
  
  
  
  
  
  
  - b) Geben Sie an, welchen Code ein RMI-Skeleton enthält! Also: Beschreiben Sie, was der Code eines Skeletons im Wesentlichen macht?