Universität Augsburg
Institut für Informatik

Deckblatt: Übung zur Vorlesung Informatik 2 SS 2018
Fakultät für Angewandte Informatik
Lehrprofessur für Informatik
Prof. Dr. Lorenz, Marius Brendle, Johannes Metzger

Hinweis: Es sind alle Felder auszufüllen! Abgabe der Übungsblätter immer **mittwochs** (Ausnahme wenn Feiertag: donnerstags) bis **spätestens 12:00 Uhr** in die entsprechend gekennzeichneten Briefkästen der Veranstaltung im Erdgeschoss des Instituts für Informatik (Gebäude N). Zuwiderhandlung wird mit Strafe geahndet! (Punktabzug)

Übungsblatt	

(hier die Nummer des bearbeiteten **Übungsblatts eintragen**)

	Übung 01 (1057 N) Montag 08:15 - 09:45 Uhr (Isabell Rücker)
	Übung 02 (1056 N) Montag 14:00 - 15:30 Uhr (Henning Cui)
	Übung 03 (1057 N) Montag 15:45 - 17:15 Uhr (Josef Kircher)
	Übung 04 (1054 N) Montag 17:30 - 19:00 Uhr (Mosaab Slimani)
	Übung 05 (1057 N) Montag 17:30 - 19:00 Uhr (David Hacker)
	Übung 06 (1055 N) Dienstag 12:15 - 13:45 Uhr (André Schweiger)
X	Übung 07 (1054 N) Dienstag 17:30 - 19:00 Uhr (Benjamin Sertolli)
	Übung 08 (1057 N) Dienstag 17:30 - 19:00 Uhr (Dat Le Thanh)
	Übung 09 (1054 N) Mittwoch 08:15 - 09:45 Uhr (Erik Pallas)
	Übung 10 (1055 N) Mittwoch 08:15 - 09:45 Uhr (Moritz Feldmann)
	Übung 11 (1054 N) Mittwoch 10:00 - 11:30 Uhr (Denise Böhm)
	Übung 12 (1056 N) Donnerstag 08:15 - 09:45 Uhr (Florian Magg)
	Übung 13 (1054 N) Donnerstag 15:45 - 17:15 Uhr (Marvin Drexelius)
	Übung 14 (1054 N) Donnerstag 17:30 - 19:00 Uhr (Patrick Eckert)
	Übung 15 (1057 N) Donnerstag 17:30 - 19:00 Uhr (Alexander Szöke)
	Übung 16 (1057 N) Freitag 08:15 - 09:45 Uhr (Philipp Braml)
	Übung 17 (1054 N) Freitag 10:00 - 11:30 Uhr (Elisabeth Korndörfer)
	Übung 18 (1054 N) Freitag 12:15 - 13:45 Uhr (Philipp Häusele)
	Übung 19 (1056 N) Freitag 12:15 - 13:45 Uhr (Maximilian Demmler)
	Übung 20 (1054 N) Freitag 14:00 - 15:30 Uhr (Florian Straßer)

(hier die eingeteilte **Übungsgruppe ankreuzen**)

Teamnummer	6

(hier die Nummer des eingeteilten **Teams eintragen**)

Tarik Selimovic
Anton Lydike
Dominic Cesnak

(hier die **Vor- und Nachnamen** aller Teammitglieder **eintragen**)

Aufgabe		
Aufgabe		
Aufgabe		
Aufgabe		
Gesamt		(vom Tutor auszufüllen)

Übungsblatt 12

45)

```
import java.awt.*;
import java.awt.event.*;

public class Aufgabe45 extends Frame implements Runnable, KeyListener {
    private static final long serialVersionUID = 1L;
    private Color color = Color.RED;
    private boolean running = false;
    private Thread thread;

    public Aufgabe25() {
        addKeyListener(this);
        setSize(100,  100);
        setBackground(this.color);
        setVisible(true);
    }

    public void toggleBackground() {
        if (color.equals(Color.RED)) {
            color = Color.GREEN;
        } else {
            color = Color.RED;
        }
        setBackground(color);
        repaint();
    }

    @Override
    public void keyTyped(KeyEvent e) {}
    @Override
    public void keyReleased(KeyEvent e) {}
    @Override
    public void keyPressed(KeyEvent e) {
        switch(e.getKeyChar()) {
            case 's':
                if (thread != null) break;
                running = true;
                thread = new Thread(this);
                thread.start();
                break;
            case 'q':
                if (thread == null) break;
                running = false;
                thread.interrupt();
                thread = null;
        }
    }

    @Override
    public void run() {
        while (running) {
            try {
                Thread.sleep(500);
                this.toggleBackground();
            } catch (InterruptedException e) {
                running = false;
            }
        }
    }
}
```

46)

```
package aufgabe46;

public class PrintThread implements Runnable {

    private boolean running = true;

    @Override
    public void run() {
        System.out.println("ausgabe");
        while (running) {
            try {
                Thread.sleep(1000);
            } catch(InterruptedException e) {
                System.out.println("Thread unterbrochen");
            }
            System.out.println("ausgabe");
        }
        System.out.println("Thread angehalten");
    }

    public void terminate() {
        running = false;
    }
}
```

47)

a)

(i)

Mögliche Ausgaben sind:

- 0
- 1
- -10
- -9

(ii)

Es können zusätzlich Checks vor dem v.print(); eingbaut werden (`t1.join(); t2.join();`), um sicherzustellen, dass beide Threads beendet wurden.

b)

Es wird immer nur `11` ausgegeben, da int ein primitiver Datentyp ist und bei der Erstellung der Threads übergeben wird. Dadurch arbeitet jeder Thread sozusagen mit seiner eigenen Variable und startet damit immer mit einer 0.

48)

a)

```
package blatt12;


import java.awt.*;
import java.awt.event.*;

public class MausFrame extends Frame implements MouseListener, MouseMotionListener {

    int width = 50;
    int height = 50;

    public MausFrame() {
        super();
        setLayout(new FlowLayout());

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
                System.exit(0);
            }
        });

        addMouseListener(this);
        addMouseMotionListener(this);

        setSize(width, height);
        setVisible(true);

    }

    public static void main(String args[]) {
        new MausFrame();
    }

    @Override
    public void mouseClicked(MouseEvent e) {
    }


    @Override
    public void mousePressed(MouseEvent e) {
    }

    @Override
    public void mouseReleased(MouseEvent e) {
    }

    @Override
    public void mouseEntered(MouseEvent e) { }

    @Override
    public void mouseExited(MouseEvent e) { }

    @Override
    public void mouseDragged(MouseEvent e) {
        int y = e.getY();
        int x = e.getX();
        this.getGraphics().clearRect(0,0, this.getWidth(), this.getHeight());
        this.getGraphics().drawLine(0,0, x, y);
        this.getGraphics().drawLine(0, this.getHeight(), x, y);
        this.getGraphics().drawLine(this.getWidth(),0, x, y);
        this.getGraphics().drawLine(this.getWidth(), this.getHeight(), x, y);
    }
```

```
    @Override
    public void mouseMoved(MouseEvent e) {

    }
}
```

b)

```
package blatt12;


import java.awt.*;
import java.awt.event.*;

public class MausFrame extends Frame implements MouseListener, MouseMotionListener {

    int width = 50;
    int height = 50;

    public MausFrame() {
        super();
        setLayout(new FlowLayout());

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
                System.exit(0);
            }
        });

        addMouseListener(this);
        addMouseMotionListener(this);

        setSize(width, height);
        setVisible(true);

    }

    public static void main(String args[]) {
        new MausFrame();
    }

    @Override
    public void mouseClicked(MouseEvent e) {
    }

    @Override
    public void mousePressed(MouseEvent e) {
    }

    @Override
    public void mouseReleased(MouseEvent e) {
    }

    @Override
    public void mouseEntered(MouseEvent e) { }

    @Override
    public void mouseExited(MouseEvent e) { }

    @Override
    public void mouseDragged(MouseEvent e) {
    }

    @Override
    public void mouseMoved(MouseEvent e) {
        this.setTitle("X: " + e.getX() + " - Y: " + e.getY());
    }
}
```