

Lösningar för tenta 1 DAT043, 2018-03-10.

Uppgift 1

```
public class SpaceShip{
    private int shields;
    private String name;

    public SpaceShip(int shields){
        this.shields = shields;
        name = "";
    }
    public int getShields(){return shields;}
    public String getName(){return name;}
    public void setName(String newName){
        name = newName;
    }
}
```

Uppgift 2

```
private static boolean ärOrdruta(char[][] ruta){
    boolean isValid = true;
    for(int i = 0; i < ruta.length; i++){
        for(int j = 0; j < ruta.length; j++){
            if(ruta[i][j] != ruta[j][i]){
                isValid = false;
            }
        }
    }
    return isValid;
}
```

Uppgift 3

Programmet skriver ut:

```
true
Yrväder
Yrväder
```

Här gäller det att hålla rätt på referensvariabler: x och y är referenser till samma array! Alltså skrivs värdet av elementet i position 1 över med "Yrväder". Variabeln b är av en primitiv typ, alltså inte en referens. Därför påverkas inte b av vad metoden gör med sin parameter r, denna är bara en annan variabel lokal till minMetod som råkar ha samma värde som b initialt.

Uppgift 4

Det saknas en ActionListener för knappen.

Lösning 1: Med lambda-uttryck. I metoden run lägger vi till raden:

```
button.addActionListener(e -> System.out.println("Löst uppgiften!"));
```

Lösning 2: Alt, men anonym klass för lyssnaren:

```
button.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
        System.out.println("Löst uppgiften!");
    }
});
```

Lösning 3: Alt, med en inre klass:

```
private class ButtonActionListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        System.out.println("Löst uppgiften!");
    }
}
```

och i metoden run() som tidigare:

```
button.addActionListener(new ButtonActionListener());
```

Uppgift 5

D – Kompileringsfel

F – Runtimefel: ClassCastException

J - Runtimefel: ClassCastException

Uppgift 6

Här ska man visa att man förstår skillnaden på klassvariabler (statiska) och instansvariabler. Det finns alltså exakt en kopia av klassvariabeln counter, och en kopia per objekt av typ Uppgift6 av instansvariabeln x. Instansvariabeln x kan inte ändras i en statisk metod, eftersom det inte finns något specifikt objekt vi kan hämta ett x från. Misstaget är alltså att metoden setVariable inte ska vara static, utan en vanlig metod. Stryker man static går klassen att kompilera.

Uppgift 7

```
public static <T> void byt(T[] arr, int x, int y){
    T temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}
```


Uppgift 9

```
public class CircularList<E>{

    private class Link {
        E elem;
        Link next = null;

        Link(E e) {
            elem = e;
        }
    }

    Link last = null;

    public void add(E elem){
        Link newLink = new Link(elem);
        if(last == null){ // tom lista.
            last = newLink;
            last.next = newLink;
        }
        else{
            newLink.next = last.next;
            last.next = newLink;
            last = newLink;
        }
    }

    public E get(int index){
        if (index < 0) {
            throw new IndexOutOfBoundsException();
        }
        // if size > the size of the circular list -1, we start over again.
        Link here = last.next; // this is the first element
        while(index > 0){
            here = here.next;
            index--;
        }
        return here.elem;
    }
}
```