Bereich: Allgemein

Zoo	**Musterlösung**
Package: `de.dhbwka.java.exercise.common.zoo`	**Klasse:** `Zoo`

```java
package de.dhbwka.java.exercise.common.zoo;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2020 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.0
 */
public class ZooAnimal {

    private String type;
    private String name;
    private String fodder;

    public ZooAnimal(String art, String name) {
        this.type = art;
        this.name = name;
    }

    /** Gibt aus, ob das Zier das übergebene Futter frisst oder verschmäht.
        Abhängig vom Vergleich mit this.futter */
    public void feed(String fodder) {
        System.out.println(String.format("%s %s %s",
              this,
              fodder.equalsIgnoreCase(this.fodder) ? "eats" : "despises",
              fodder));
    }

    @Override
    /** Informationen zu einem ZooAnimal */
    public String toString() {
        return name + " (" + type + ")";
    }

      /**
       * @return the type
       */
      public String getType() {
            return type;
      }

      /**
       * @param type the type to set
       */
      public void setType(String type) {
            this.type = type;
      }
```

```
	/**
	 * @return the name
	 */
	public String getName() {
		return name;
	}

	/**
	 * @param name the name to set
	 */
	public void setName(String name) {
		this.name = name;
	}

	/**
	 * @return the fodder
	 */
	public String getFodder() {
		return fodder;
	}

	/**
	 * @param fodder the fodder to set
	 */
	public void setFodder(String fodder) {
		this.fodder = fodder;
	}

}
```

```
package de.dhbwka.java.exercise.common.zoo;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2020 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.0
 */
public class Predator extends ZooAnimal {

    public Predator(String type, String name) {
        super(type, name);
        setFodder("flesh");
    }

}
```

```java
package de.dhbwka.java.exercise.common.zoo;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2020 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.0
 */
public class Songbird extends ZooAnimal {

    public Songbird(String art, String name) {
        super(art, name);
        setFodder("grains");
    }

}
```

```java
package de.dhbwka.java.exercise.common.zoo;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2020 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.0
 */
@SuppressWarnings("serial")
public class ZooCapacityException extends Exception {

    public ZooCapacityException() { }

    public ZooCapacityException(String msg) {
        super(msg);
    }

}
```

```java
package de.dhbwka.java.exercise.common.zoo;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2020 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.0
 */

@SuppressWarnings("serial")
public class ZooFileException extends Exception {

    public ZooFileException() { }

    public ZooFileException(String msg) {
        super(msg);
    }

}
```

```java
package de.dhbwka.java.exercise.common.zoo;

import java.io.FileWriter;
import java.io.PrintWriter;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2020 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.0
 */
public class Zoo {

    public static final int STANDARD_CAPACITY = 5;
    public final int MAX;           // Maximalzahl der Tiere im Zoo
    public ZooAnimal animals[];     // Alle Tiere dieses Zoos
    public int noOfAnimals = 0;     // Anzahl der Tiere im Zoo

    public Zoo() {
        this(STANDARD_CAPACITY);
    }

    /** Erzeugt einen neuen Zoo mit der Kapazität für MAX Tiere */
    public Zoo(int MAX) {
        this.MAX = MAX;
        animals = new ZooAnimal[MAX];
    }

    /** Fügt, weitere Kapazität vorausgesetzt, dem Zoo ein Tier hinzu */
    public void addAnimal(ZooAnimal animal) throws ZooCapacityException {
```

```
        if (noOfAnimals<MAX) {
            animals[noOfAnimals++] = animal;
            System.out.println(animal.toString() + " added to zoo");
        } else
            throw new ZooCapacityException("Error: zoo capacity exceeded!");
    }

    /** Testet, ob im Zoo ein Tier mit dem angegebenen Namen existiert */
    public boolean existsAnimal(String name) {
        for (ZooAnimal animal : animals)
            if (animal!=null && animal.getName().equalsIgnoreCase(name))
                return true;
        return false;
    }

    /** Liefert alle Tiere des Zoos in einem passend großen Array */
    public ZooAnimal[] getAnimals() {
        ZooAnimal[] result = new ZooAnimal[noOfAnimals];
        for (int i=0; i<noOfAnimals; i++)
            result[i] = animals[i];
        return result;
    }

    /** Speichert alle Tiere des Zoos in einer Datei */
    public void saveToFile(String filename) throws ZooFileException {
        try (PrintWriter pw = new PrintWriter(new FileWriter(filename))) {
            for (ZooAnimal animal : this.getAnimals())
                pw.println(animal.getType()+";"
                   +animal.getName()+";"
                   +animal.getClass().getSimpleName());
        } catch (Exception ex) {
            throw new ZooFileException("Error saving file.");
        }
    }

    /** Füttert alle Tiere des Zoos mit dem übergebenen Futter */
    public void fuettern(String futter) {
        for (ZooAnimal animal : this.getAnimals())
            animal.feed(futter);
    }

    public static void main(String[] args) {
        int capacity;
        try {
            capacity = Integer.parseInt(args[0]);
        } catch (Exception e) {
            capacity = STANDARD_CAPACITY;
        }
        Zoo z = new Zoo(capacity);
        /* add ZooAnimals */
        try {
            z.addAnimal(new Predator("Tiger", "Fred"));
            z.addAnimal(new Predator("Tiger", "Lisa"));
            z.addAnimal(new Predator("Lion", "Simba"));
            z.addAnimal(new Songbird("Nuthatch", "Hansi"));
            z.addAnimal(new Songbird("Blackbird", "Sina"));
            z.addAnimal(new Songbird("Wren", "Henry"));
        } catch (ZooCapacityException ex) {
```

```
            System.err.println(ex.getMessage());
        }
        // Save animals to file
        try {
            z.saveToFile("ZooAnimals.txt");
            System.out.println("Saved animals to file!");
        } catch (ZooFileException ex) {
            System.err.println(ex.getMessage());
        }
        // Feed the animals
        z.fuettern("grains");
        // Schauen, ob verschiedene Tiere im Zoo vorkommen
        String[] names = {"Fred","Kimba","Henry","Lotte"};
        for (String name : names)
            System.out.println("Is '"+name+"' in the zoo? "
             + (z.existsAnimal(name) ? "yes" : "no"));
    }

}
```