

Zusatzaufgaben OMP – Cupboard

Gegeben sei folgender Code:

```
public class Cupboard<T extends PhysicalObject> {
    private boolean[][][] room;
    private Volume volume;
    private List<T> content = new ArrayList<>();

    public Cupboard(Volume volume){
        this.volume = volume;
        room = new
boolean[volume.getWidth()][volume.getHeight()][volume.getLength()];
    }
}

abstract class PhysicalObject {
    private Volume volume;

    public PhysicalObject(Volume volume){
        this.volume = volume;
    }

    Volume getVolume(){return volume;};
}

public class Book extends PhysicalObject {
    private String name;
    private String author;
    private String isbn;

    public Book(Volume volume, String name, String author, String isbn) {
        super(volume);
        this.name = name;
        this.author = author;
        this.isbn = isbn;
    }

    @Override
    public Volume getVolume() {
        return null;
    }

    public String getName() {
        return name;
    }
}
```

```

    }

    public String getAuthor() {
        return author;
    }

    public String getIsbn() {
        return isbn;
    }
}

public class Volume {
    private int width;
    private int height;
    private int length;

    public Volume(int width, int height, int length){
        this.width = width;
        this.height = height;
        this.length = length;
    }

    public int getWidth() {
        return width;
    }

    public int getHeight() {
        return height;
    }

    public int getLength() {
        return length;
    }
}

```

Erweitern sie die Cupboard-Klasse um Methoden zum Finden und entfernen von Objekten: boolean contains(T obj), boolean remove(T obj). Weiterhin soll der Index des eines Objektes über eine Methode int getIndexOf(T obj), welche eine Exception wirft, sollte das Objekt nicht zu finden sein. Außerdem sollen alle weiteren Methoden, ggf. auch neue Klassen, implementiert werden, die zu Ausführung folgender Main-Methode notwendig sind. Die Ausführung der Methode soll folgende Konsolenausgabe verursachen:

```

public class CupboardTest {
    public static void main(String[] args){
        Cupboard<Book> boardOne = new Cupboard<>(new Volume(30, 150, 15));
        try{

```

```

        Book lotr = new Book(new Volume(5, 20, 14), "The Lord of the Rings",
"J.R.R.Tolkien", "9780395595114")
        boardOne.store(lotr);
        boardOne.store(new Book(new Volume(2, 20, 14),"Psychopathen", "Kevin
Dutton", "9780374291358"));
        boardOne.store(new Book(new Volume(30, 150, 15), "Pile of Paper", "John
Doe", "-----"));
        boardOne.store(new Book(new Volume(3, 17, 12), "Also Sprach
Zarathustra", "Friedrich Nietzsche", "9788408010258"));
        boardOne.remove(lotr);
    }catch(Exception e){
        System.out.println(e.toString());
    }
    boardOne.displayContent();
    Cupboard<Games> boardTwo = new Cupboard<>(new Volume(10, 20, 10));
    try{
        int counter = 0;
        while(true){
            boardTwo.store(new Game("Game No " + counter, "John Doe"));
        }
    }catch(Exception e){
        System.out.println(e.toString());
    }
    boardTwo.displayContent();
}
}

```

Konsolenausgabe:

Es ist ein Fehler aufgetreten. Das Objekt war zu groß.

Inhalt:

The Lord of the Rings, J.R.R.Tolkien, isbn: 9780395595114

Psychopathen, Kevin Dutton, isbn: 9780374291358

Empty room: 65540

Es ist ein Fehler aufgetreten. Das Objekt war zu groß.

Game No 1, John Doe

Game No 2, John Doe

Game No 3, John Doe

Game No 4, John Doe

Game No 5, John Doe

Game No 6, John Doe

Game No 7, John Doe

Game No 8, John Doe

Game No 9, John Doe

Game No 10, John Doe

Empty room: 0