Peer review t1

First peer review with CornerNerds()!
The biggest learnings for this round of peer review were that setters can be used instead of a constructor.
By doing so, it is possible to create an empty instance of a class. The properties will get set later on. This could be useful if we don't know the data that is supposed to be used as a property.
For exercise 5 we discussed why the ShoeCostumer class is a data class and not a service class. What confused us, was that there is a method inside of the ShoeCostumer class, but this method only compares two values and returns a boolean. By doing so it provides data and therefore the ShoeCostumer class is a data class and not a service class. It only provides data and isn't manipulating data.

```
public class ShoeCustomer {
    2 usages
    private int shoeSize;

    2 usages
    public ShoeCustomer(int shoeSize) {
        this.shoeSize = shoeSize;
    }

    1 usage
    public boolean tryShoe(Shoe shoe) { // Checks if the sizes are matching. This method can't be static!
        return shoe.getSize() == shoeSize;
    }
}
```

We also stumbled over the static keyword. What we found is that by making a method static, it then can be used without creating an instance of the class.
But a static function can only be used with static variables and also can only call static functions itself.
Static can only be used if there is only one instance of the class, and it should be a service class.

```
public class Application {
    public static void main(String[] args) {
        ShoeCustomer matilda = new ShoeCustomer( shoeSize: 38); // Create two customers.
        ShoeCustomer hansel = new ShoeCustomer( shoeSize: 42);
        // Use the method findShoeForCustomer() of class ShopAssistant to check the shoe sizes. The method is static.
        ShopAssistant.findShoeForCustomer(matilda);
        ShopAssistant.findShoeForCustomer(hansel);
    }
}
```