



Object Oriented Programming 2

Sequence Diagrams, Collections, Enums & Static • Week 3

IDOARRT

- **Intention**

- Get to know the sequence diagram and why we create this;
- Get to know different collections, what is enum and what is static;

- **Desired Outcome**

- You know what a sequencediagram is and are able to create one;
- You know different type of collections, enum and static and you can use that in your code.

- **Time**

45 minutes.

IDOARRT • Agenda

- Wha Sequence Diagram?
- How does a Sequence Diagram look like?
- Collections
 - HashSet
 - HashMap
- Enums
- Static



Design

Sequence Diagram

Why aSequence Diagram?

- Visualisation of interactions
 - Shows how objects communicate to execute a dedicated method.
- Can be connected to the Classdiagram

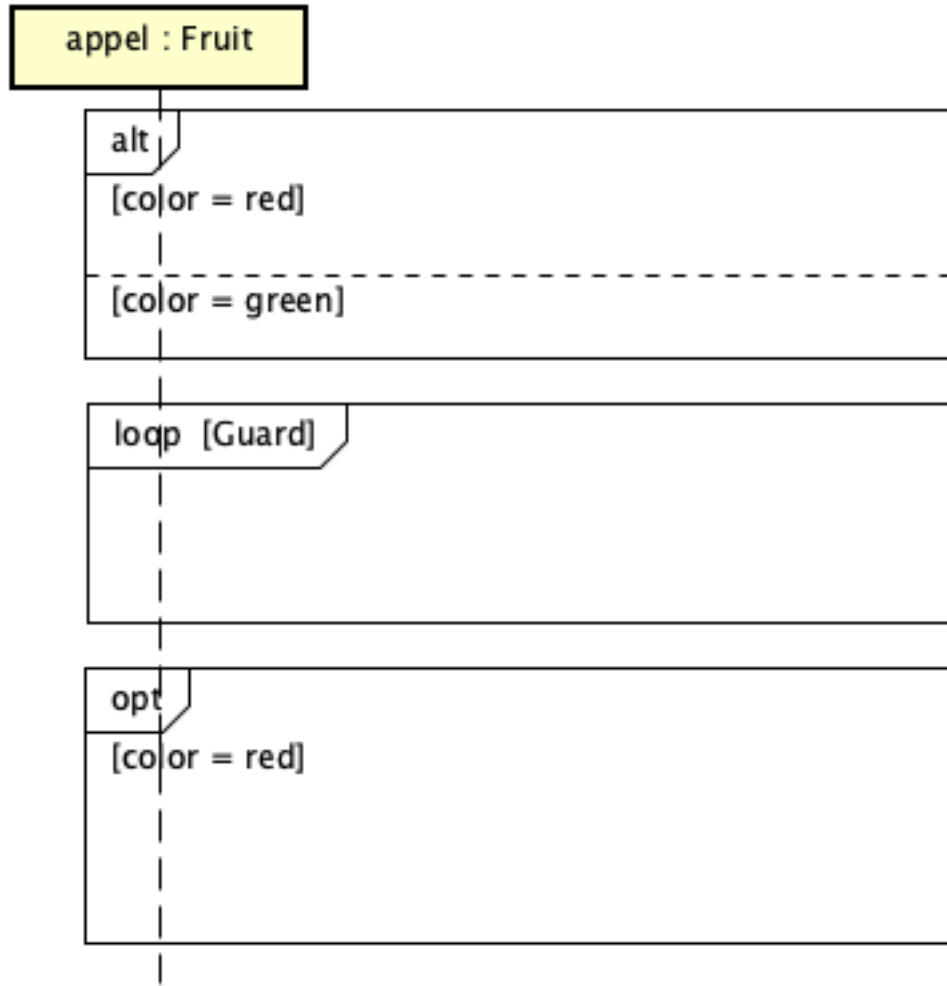
De components: Lifeline

appel : Fruit

A UML Lifeline diagram showing a single lifeline for the object 'appel' of type 'Fruit'. The lifeline is represented by a vertical dashed line extending downwards from a yellow rectangular box containing the text 'appel : Fruit'.

- **Lifeline**
Individual component that participates in an action
- Lifeline of the object apple of the Class Fruit.

De componenten: Frame

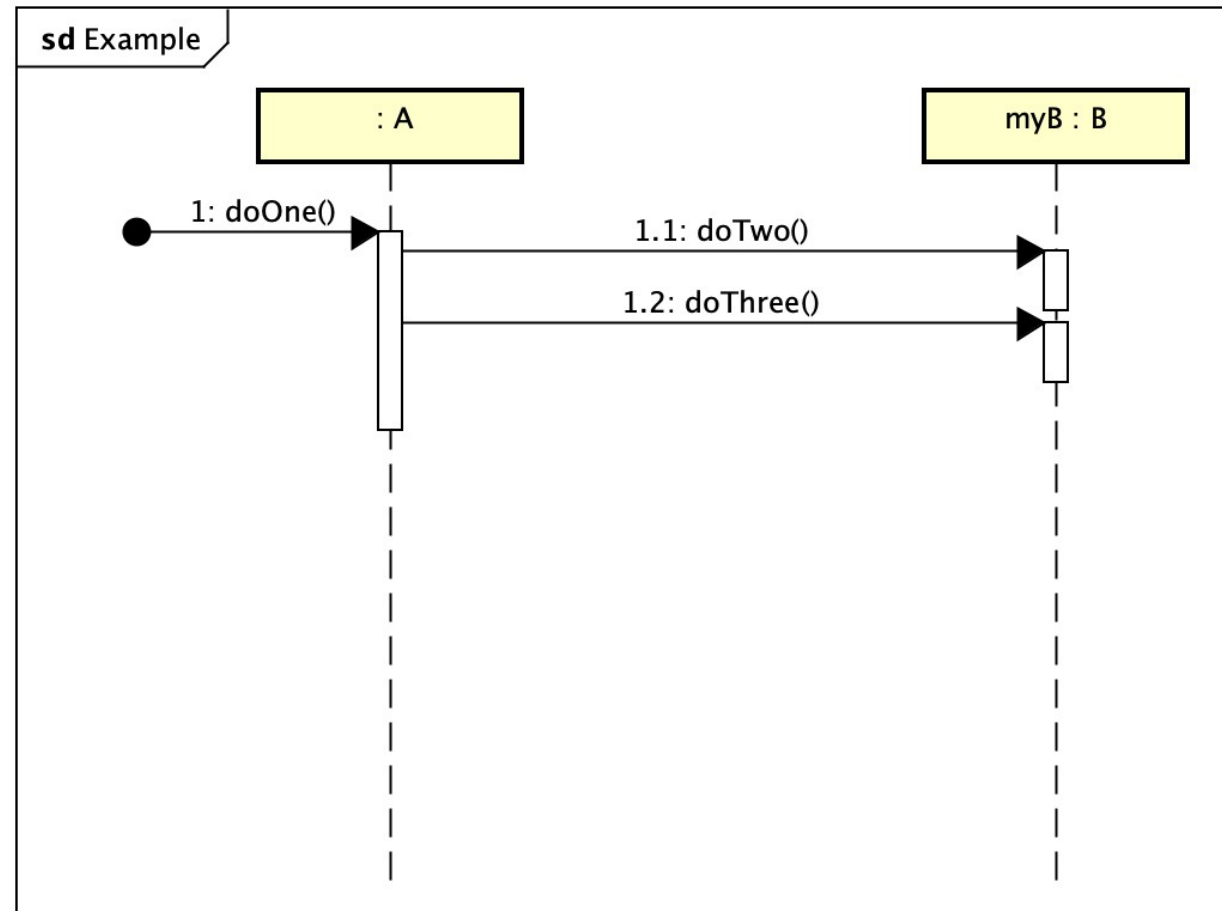


- **Frame**
For conditions and loops, these can be nested.
- 1. Multiple if's / switch case
- 2. Loop, eg. foreach
- 3. Single if

Example

```
public class A
{
    private B myB = new
    B();

    public void doOne()
    {
        myB.doTwo();
        myB.doThree();
    }
}
```





Realise

Collections, Enum, Static

Collections • HashSet

- Can contain every element only once

```
1 User user = new User("Jan");
2
3 HashSet<User> users = new HashSet<>();
4
5 users.add(user);
6
7 users.size(); // Returns 1
8
9 users.add(user);
10
11 users.size(); // Returns 1
```

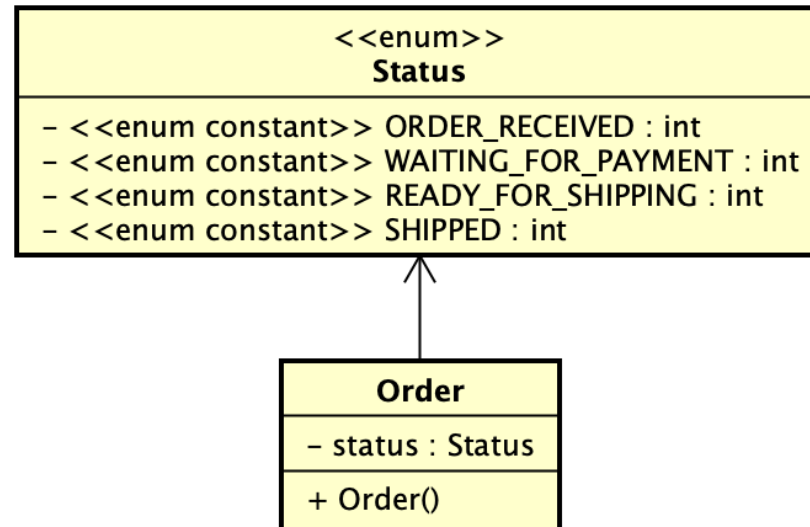
Collections • HashMap

- Key/value pair

```
1 HashMap<String, String> phonebook = new HashMap<String, String>();  
2  
3 phonebook.put("Jan", "+31612345678");  
4  
5 phonebook.get("Jan"); // Returns +31612345678  
6  
7 phonebook.put("Jan", "+31687654321");  
8  
9 phonebook.get("Jan"); // Returns +31687654321
```

- Key can be every object type
 - Only objects, so int becomes Integer

Enum in the classdiagram



Enums

- Pre-defined values

```
1 enum Status
2 {
3     ORDER_RECEIVED,
4     WAITING_FOR_PAYMENT,
5     READY_FOR_SHIPPING
6     SHIPPED
7 }
```

```
1 public class Order
2 {
3     private Status status;
4
5     public void setStatus(Status status)
6     {
7         this.status = status;
8     }
9 }
```

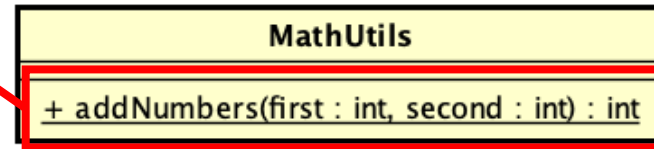
```
1 Order order = new Order();
2 order.setStatus(Status.READY_FOR_SHIPPING);
```

Static

- Fields and methods
 - Shared between objects
- Call without instantiating

Static in the classdiagram

Underlined.



Static in code

```
1 public class MathUtils
2 {
3     public static int addNumbers(int first, int second)
4     {
5         return first + second;
6     }
7 }
```

```
1 MathUtils utils = new MathUtils();
2 utils.addNumbers(1, 2); // Doesn't work
3
4 MathUtils.addNumbers(1, 2); // Returns 3
```


Goodluck!