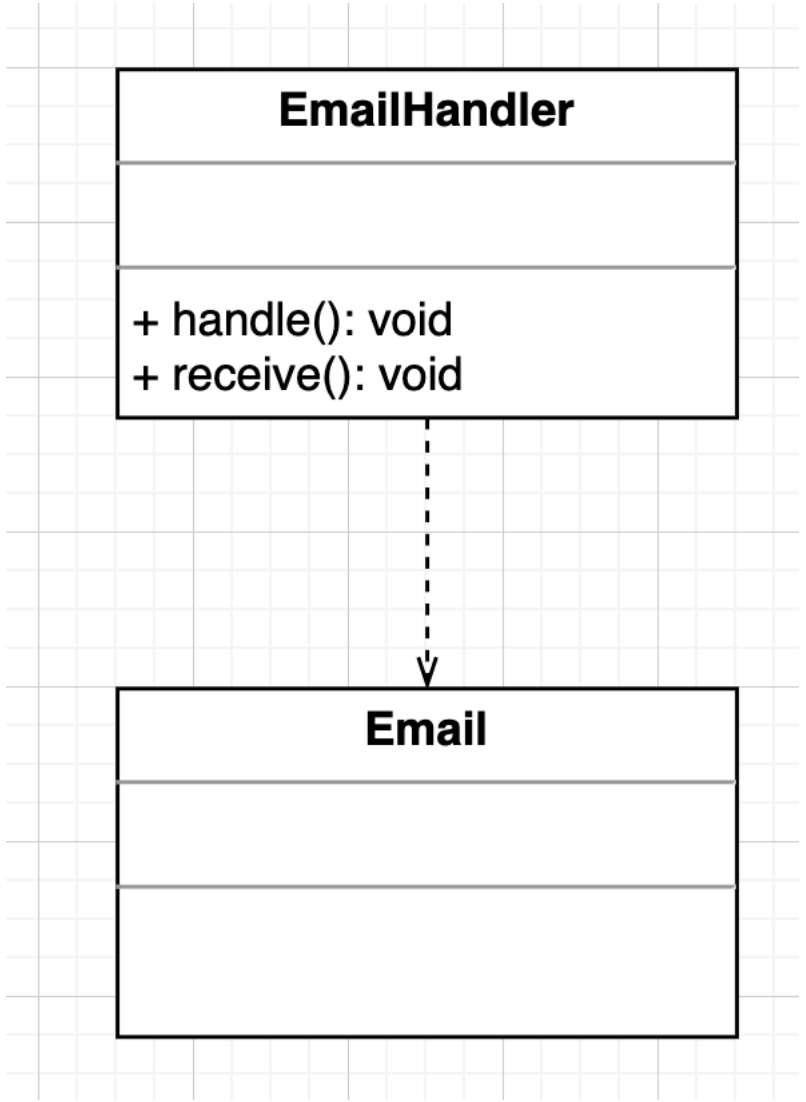


# EmailHandler

Group 2

An **Email Handler** for enterprise has ability to **handle** all **received emails**.



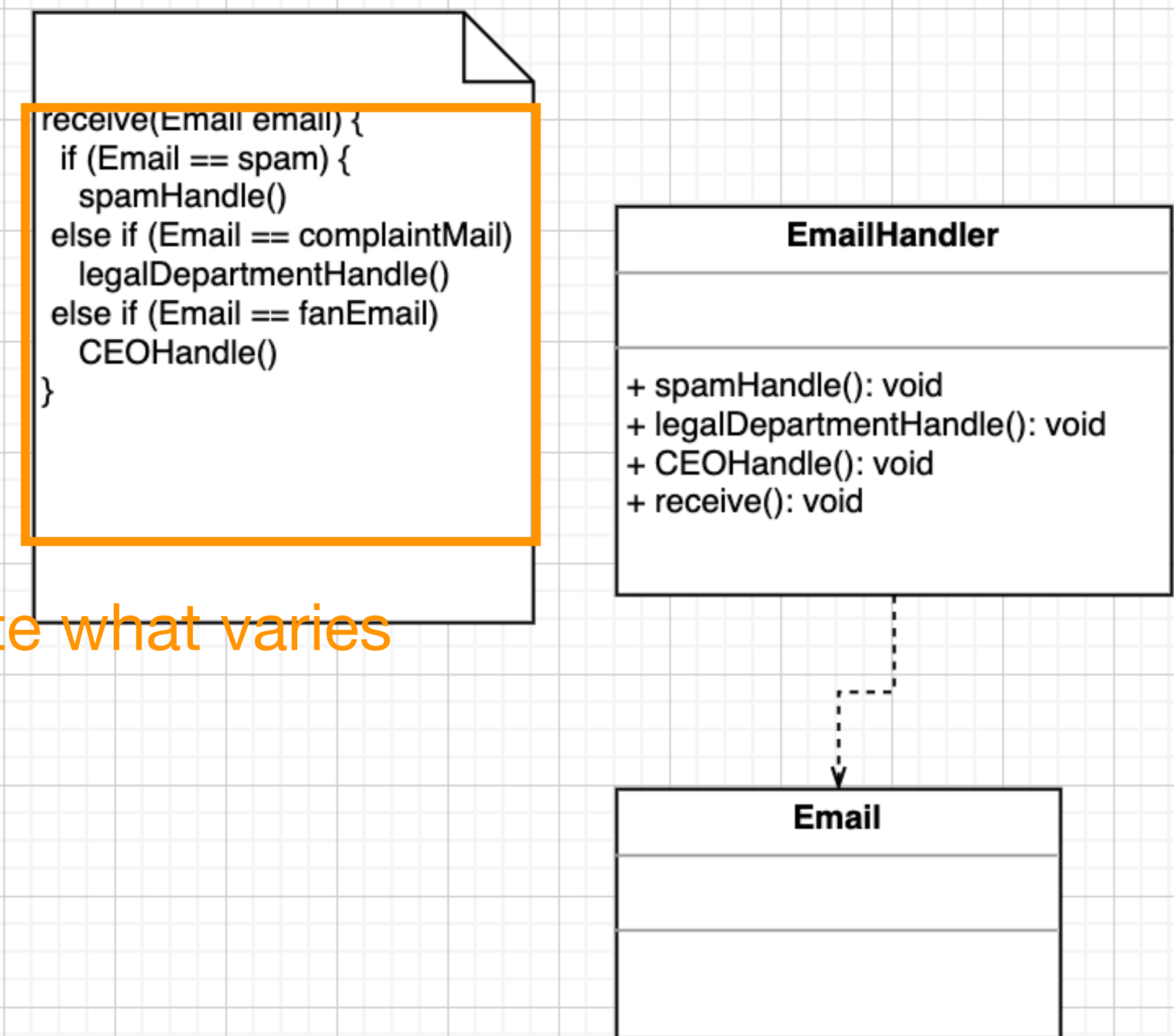
❑ The mail handling process of the Email Handler is as follows:

If an email is a spam, it will be **put** in a **spam box**.

If an email is a complaint mail rather than a spam, it will be **forwarded** to the **legal department**.

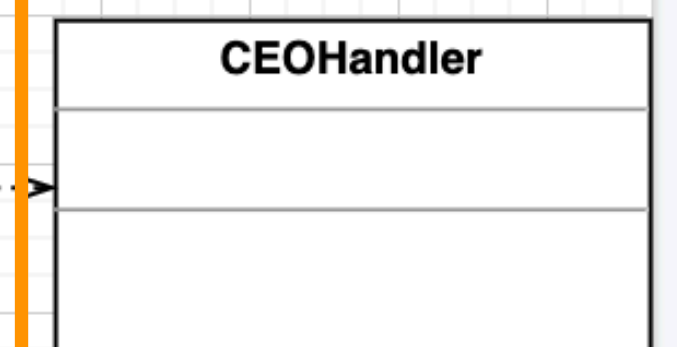
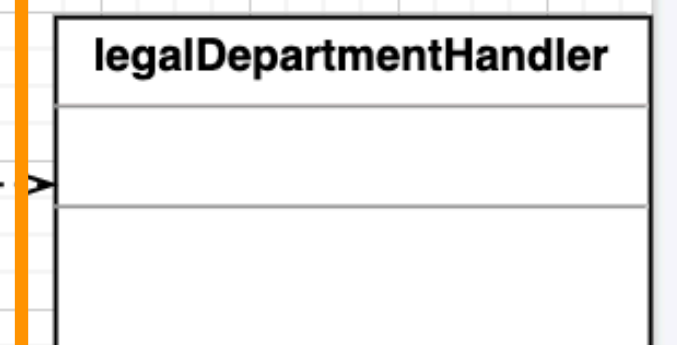
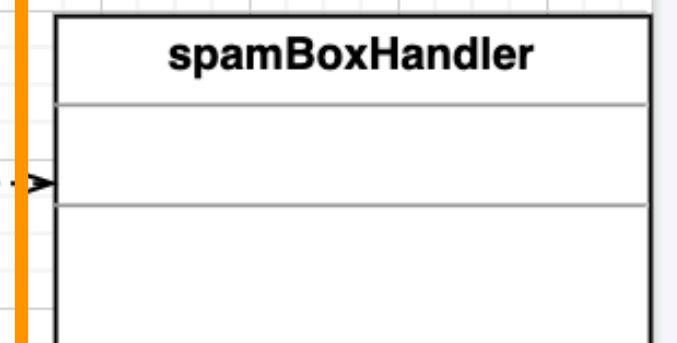
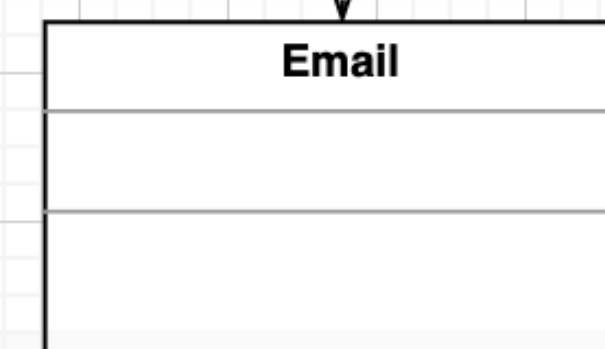
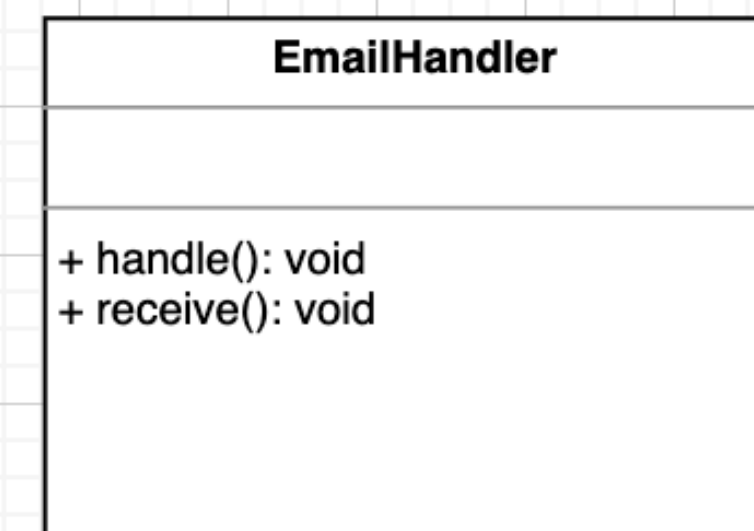
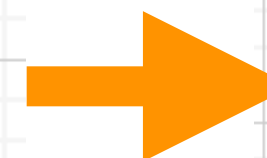
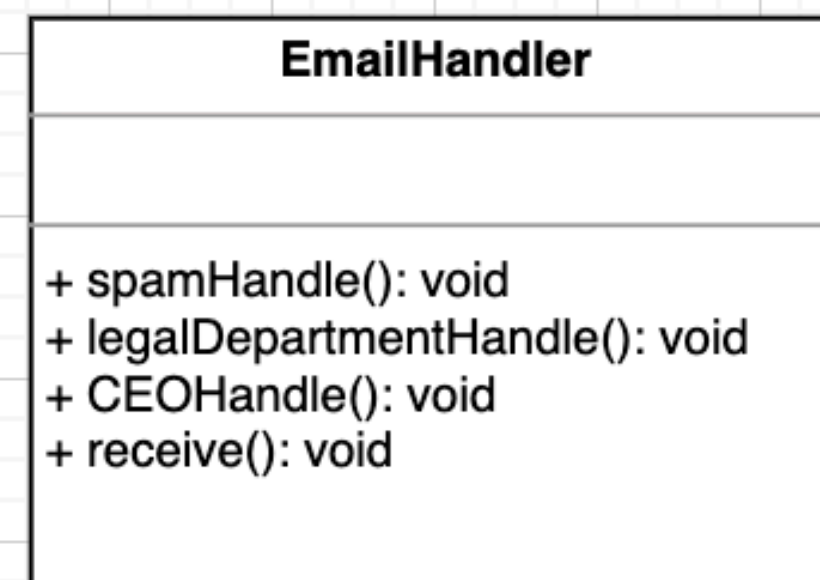
If an email is a fan email, it will be **forwarded** to the **CEO**.

encapsulate what varies

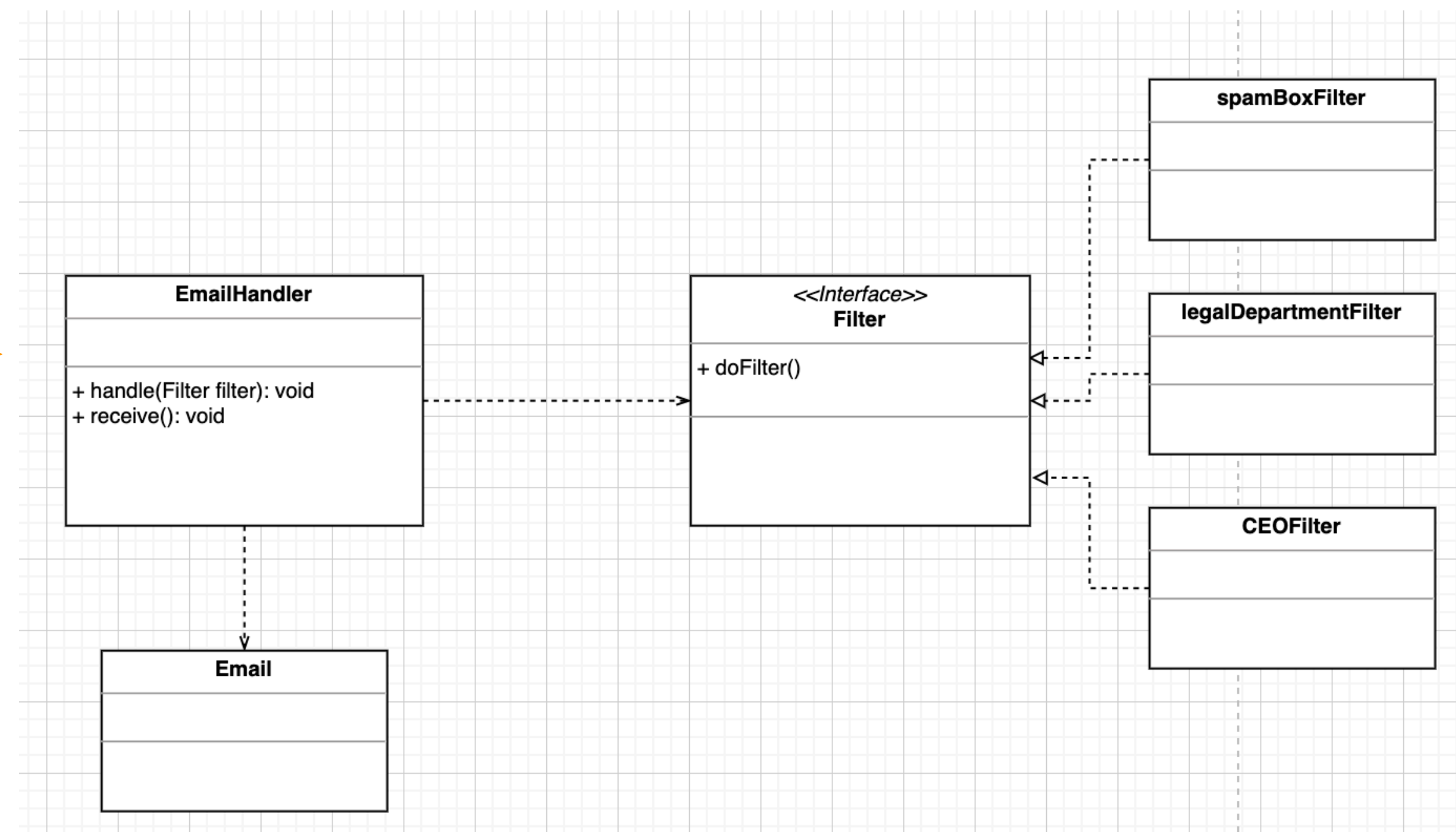
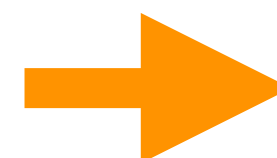
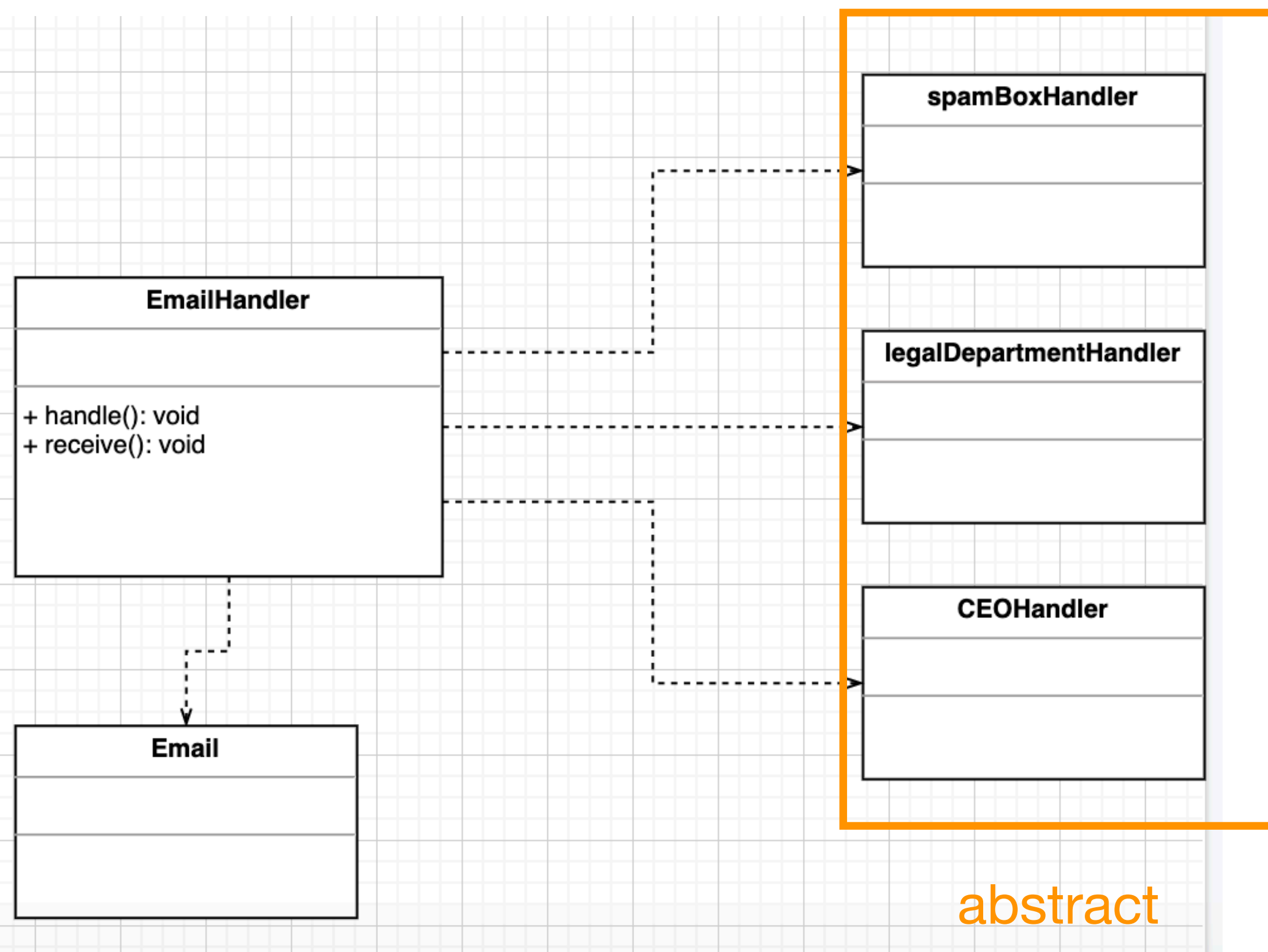


```
receive(Email email) {  
    if (Email == spam) {  
        spamHandle()  
    }  
    else if (Email == complaintMail)  
        legalDepartmentHandle()  
    else if (Email == fanEmail)  
        CEOHandle()  
}
```

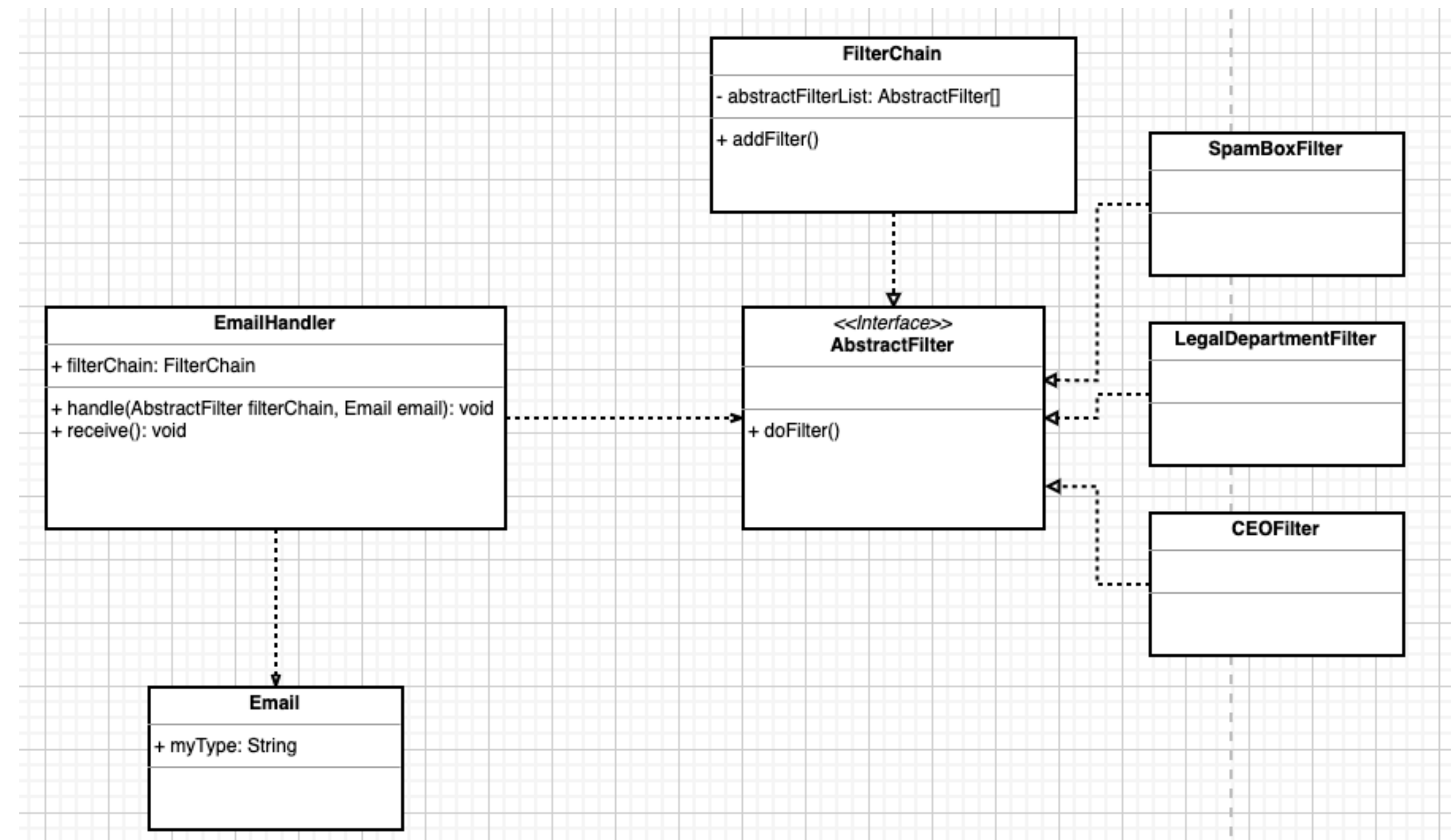
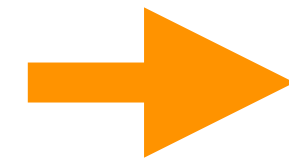
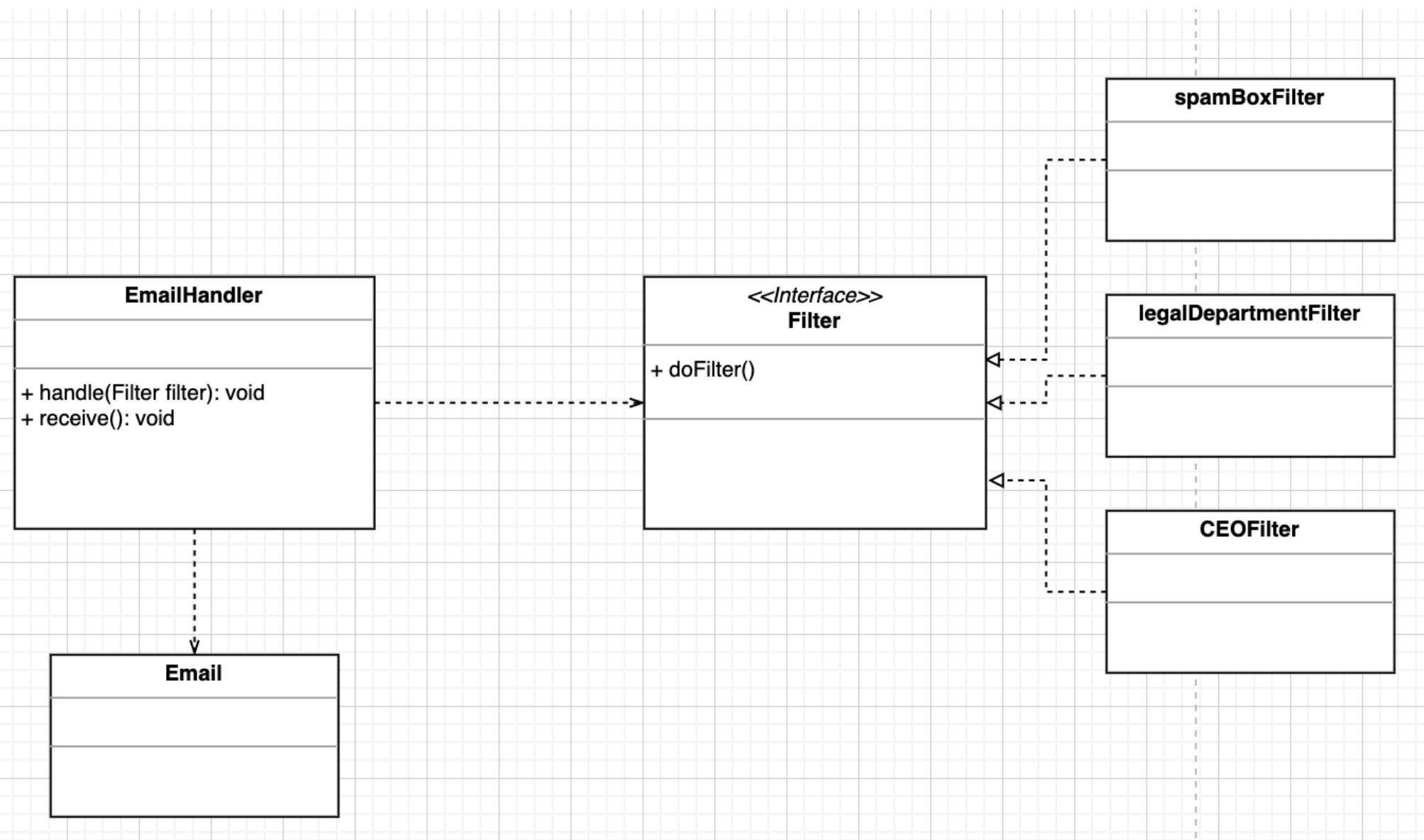
encapsulate what varies



abstract



# Order many filters with FilterChain



```
public class FilterChain implements AbstractFilter {
    private int pos = 0;
    public void setPos(int pos) {
        this.pos = pos;
    }
    private List<AbstractFilter> abstractFilterList = new ArrayList<>();
    @Override
    public void doFilter(AbstractFilter abstractFilter, Email email) {
        abstractFilterList.get(pos++).doFilter(this, email);
    }
    public void addFilter(AbstractFilter abstractFilter) {
        abstractFilterList.add(abstractFilter);
    }
}
```

```
public class SpamBoxFilter implements AbstractFilter{
    @Override
    public void doFilter(AbstractFilter filterChain, Email email) {
        if (email.getMyType().equals("SPAM")) {
            System.out.println("Put mail to the spam box.");
            return;
        }
        filterChain.doFilter(filterChain, email);
    }
}
```

```
public class EmailHandler {
    public FilterChain filterChain;

    public EmailHandler(FilterChain filterChain) {
        this.filterChain = filterChain;
    }

    public void handle(Email email) {
        this.filterChain.doFilter(filterChain, email);
    }
}
```

```
public interface AbstractFilter {
    public void doFilter(AbstractFilter filter, Email email);
}
```

