What Is Aspect Oriented Programming?



Cătălin Tudose
PHD IN COMPUTER SCIENCE, JAVA AND WEB TECHNOLOGIES EXPERT
www.linkedin.com/in/catalin-tudose-847667a1

Overview



Introducing Aspect Oriented Programming

Common AOP use cases

Aspects, join points, advices, pointcuts, target objects, introductions

AOP proxy and weaving



A method trying to get information about a passenger:

```
public class PassengerService {
  public static Passenger getPassenger(int id) {
    return Database.getPassenger(id);
  }
}
```



```
public static Passenger getPassenger(int id) {
  log.debug("Call method getPassenger with id " + id);
  Passenger passenger = Database.getPassenger(id);
  log.debug("Passenger info: " + passenger.toString());
  return passenger;
}
```



```
public static Passenger getPassenger(int id) {
 log.debug("Call method getPassenger with id " + id);
 Passenger passenger;
 try {
   passenger = Database.getPassenger(id);
 } catch(SQLException e) {
   throw new RuntimeException(e);
 log.debug("Passenger info: " + passenger.toString());
 return passenger;
```

```
public static Passenger getPassenger(int id) {
 if (!SecurityContext.hasRights("getPassenger")) {
    throw new RuntimeException("Permission denied!");
 log.debug("Call method getPassenger with id " + id);
 Passenger passenger;
 try {
    passenger = Database.getPassenger(id);
  } catch(SQLException e) {
    throw new RuntimeException(e);
  log.debug("Passenger info: " + passenger.toString());
  return passenger;
```

```
public static Passenger getPassenger(int id) {
  if (!SecurityContext.hasRights("getPassenger")) {
    throw new RuntimeException("Permission denied!");
  log.debug("Call method getPassenger with id " + id);
  Passenger passenger;
  try {
    if (cache.contains(id)) {
        passenger = cache.get(id);
    } else {
      passenger = Database.getPassenger(id);
      cache.put(id, passenger);
  } catch(SQLException e) {
    throw new RuntimeException(e);
  log.debug("Passenger info: " + passenger.toString());
  return passenger:
```

How Did It Evolve?

Large amount of service code

17 lines instead of one, and the code continues to grow



Common Use Cases of AOP

Logging

User rights check

Transactions

Exception handling

Caching

Tweaking a legacy application



Aspects

Do something before the method

Call the method

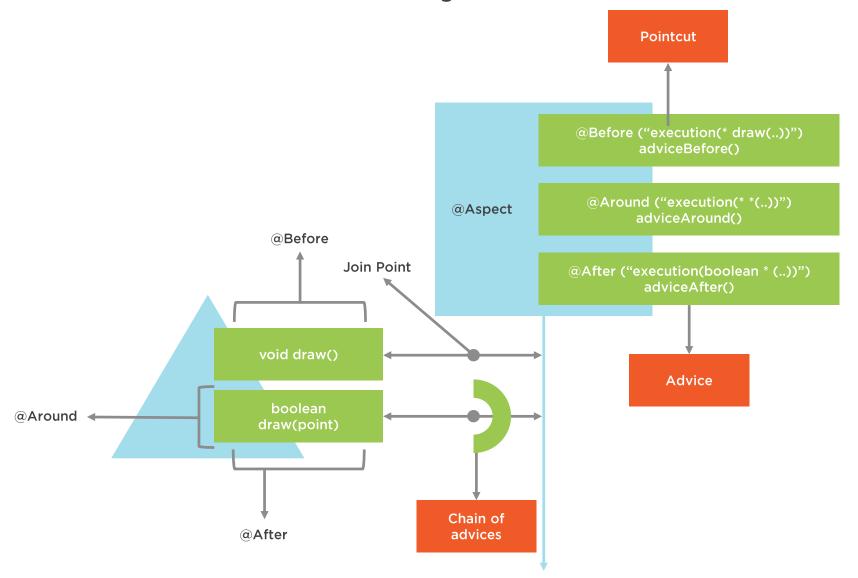
Do something after the method





```
public static Passenger getPassenger(int id) {
  log.debug("Call method getPassenger with id " + id); → @Before advice
  Passenger passenger = Database.getPassenger(id);
  log.debug("Passenger info: " + passenger.toString()); → @After advice
  return passenger;
}
```



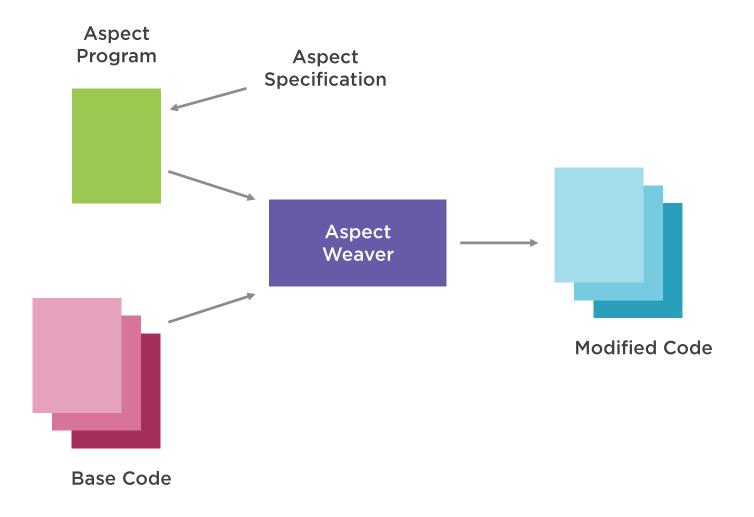




Target object Introduction Weaving **AOP Proxy**



AOP Proxy and Weaving





Summary



The need for Aspect Oriented Programming

AOP use cases for orthogonal functionality

Aspects, join points, advices, pointcuts, target objects, introductions

AOP proxy and weaving

