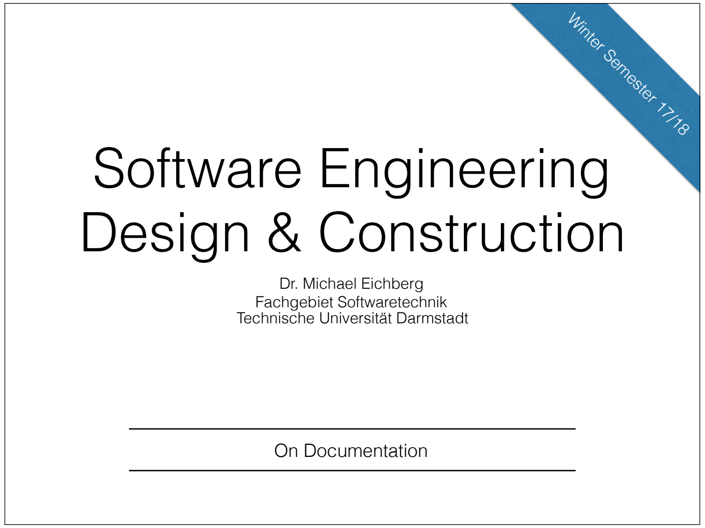
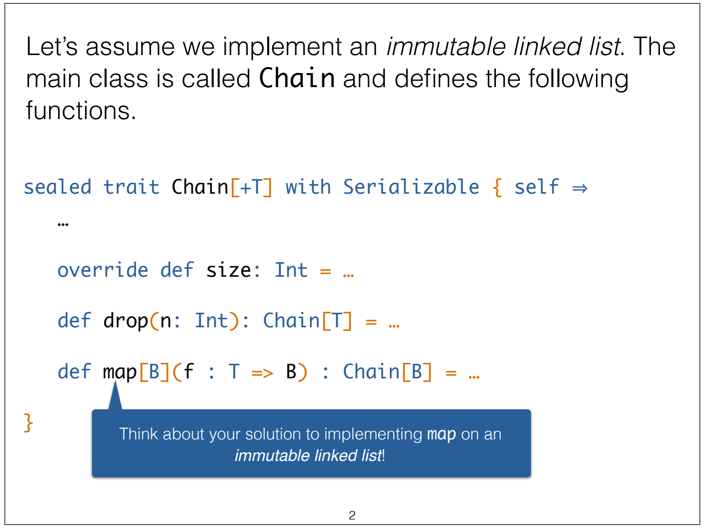


Winter Semester 17/18

Software Engineering Design & Construction

Dr. Michael Eichberg
Fachgebiet Softwaretechnik
Technische Universität Darmstadt

On Documentation



Let's assume we implement an *immutable linked list*. The main class is called `Chain` and defines the following functions.

```
sealed trait Chain[+T] with Serializable { self ⇒
   …
   override def size: Int = …

   def drop(n: Int): Chain[T] = …

   def map[B](f : T => B) : Chain[B] = …
}
```

Think about your solution to implementing `map` on an *immutable linked list*!

In case of size we definitively want to document the complexity as hint to developer that – if size information is often required – this operation is potentially expensive!

In case of drop, we want to document the general behavior (drops the number of specified elements) and also its behavior in case the chain has less than the number of elements - here, we have (at least) two options: throw an exception or just ignore!

When you document the function think about the following: a naive implementation of `map` for *immutable linked lists* would revert the order of the elements which would violate the expectation of users. An efficient implementation that does not violate the expectation is possible!