

Aufgabe ListServ: Methoden höherer Ordnung, Pattern Matching für Listen

Gegeben sind der Scala-Trait `ListService` und der Testtreiber `ListServiceTest`.

```
/**Trait for List operations.
 * A Scala trait is much like a Java interface, but can contain concrete methods.
 * You can implement it either by extending or by withing it. For this exercise you should write:
 * class ListServiceImpl extends ListService { ... }
 * Generic type parameters are notated in Scala by square brackets []
 */
trait ListService {

  /**Merges the two sorted Lists a and b into a sorted result List.
   * Sorting is done according to the predicate.
   * @param before A predicate judging two List elements, if the left one should be listed before the right one.
   * @throws IllegalArgumentException List a or List b is not sorted according to the predicate.*/
  def mergeSorted[T](a: List[T], b: List[T], before: (T,T)=>Boolean): List[T]

}
```

Implementieren Sie diesen Trait in einer Klasse `ListServiceImpl`.

Testen Sie ihn mittels des mitgelieferten Testtreibers.

Hinweise

Bei der Untersuchung der Listen `a` und `b` sollen Sie das Abspalten des Listenkopfes und -Restes mit Hilfe der **match**-Anweisung vornehmen.