

## 5 Abstracting with Function Objects

Download the file `wed-pm-lab-part2.java`.

We will now practice the use of *function objects*. The only purpose for defining the class `SmallImageFile` is to implement one method that determines whether the given `ImageFile` object has the desired property. An instance of this class can then be used as an argument to a method that deals with `ImageFiles`.

1. In the `Examples` class design the tests for the class `SmallImageFile`.
2. Design the method `allSmallerThan40000` that determines whether all items in a list are smaller than 40000 pixels. The method should take an instance of the class `SmallImageFile` as an argument.
3. Design the class `NameShorterThan4` that implements the `ISelect` interface with a method that determines whether the name in the given `ImageFile` object is shorter than 4.  
Make sure in the class `Examples` you define an instance of this class and test the method.
4. Design the method `allNamesShorterThan4` that determines whether all items in a list have a name that is shorter than 4 characters. The method should take an instance of the class `NameShorterThan4` as an argument.
5. Design the method `allSuch` that determines whether all items in a list satisfy the predicate defined by the `select` method of a given instance of the type `ISelect`. In the `Examples` class test this method by abstracting over the method `allSmallerThan40000` and the method `allNamesShorterThan4`.
6. Follow the same steps as above to design the method `anySuch` that determines whether there is an item a list that satisfies the predicate defined by the `select` method of a given instance of the type `ISelect`.