

Homework 10

Please email your solutions to Rongdi Huang (rh1424@nyu.edu). Solutions to programming exercises **must** be submitted electronically as plain text files. No exotic formats, please!

The deadline for Homework 10 is December 13, 8pm.

Problem 1 Generics and Variance (10 Points)

1. Write a generic Scala method `middle` that returns the middle element from any `Iterable[T]`. For example, `middle("World")` is `'r'`. **(3 Points)**
2. The Scala compiler maps the type `Int` to the Java type `Int`. In particular, this means that the Scala type `Int` also implements the Java interface `Comparable[Int]`. The Scala standard API then defines a type `RichInt`, which provides additional operations on integers using implicit conversions (following the *Pimp my Library* pattern we have seen in class). If you inspect the API documentation you will see that the type `RichInt` implements the interface `Comparable[Int]` rather than `Comparable[RichInt]`. What is the reason for this? **(2 Points)**
3. Consider the following two generic classes representing immutable and mutable pairs of values:

```
class Pair[+T,+S](val fst: T, val snd: S) {
  def replaceFst[U >: T](newFst: U): Pair[U,S] =
    new Pair(newFst, snd)
  def replaceSnd[U >: S](newSnd: U): Pair[T,U] =
    new Pair(fst, newSnd)
}

class PairM[T,S](private[this] var _fst: T,
                 private[this] var _snd: S) {
  def fst = _fst
  def snd = _snd
  def replaceFst(newFst: T) { _fst = newFst }
  def replaceSnd(newSnd: S) { _snd = newSnd }
}
```

Define a generic method `swap` that takes an immutable pair and returns a new pair with the components swapped. Write a similar method for mutable pairs that performs the swap in-place instead of allocating a new object. How do the two methods differ in the type of their parameters (other than that the first one takes a `Pair` and the second a `PairM`)? **(3 Points)**

4. The type parameters `T` and `S` of class `PairM` cannot be declared covariant. If the class was covariant, we could construct a similar counterexample for type safety as for the class `Cell`. The parameters can also not be declared contravariant. Why is that? Construct an example of the problem. **(2 Points)**