

Imię i nazwisko	Data	Godzina
Mateusz Grzesiuk	15.12.2020r.	13:15

	1	2	3	4	5	6	7	8	9	10
nr listy: 9 zrobione	V	V	V	V						
nr listy: 10 zrobione	V	V	V	V						

Lista 10

Zadanie 1

- a) Parametr typu danej klasy może być kowariantny tylko wówczas, gdy jest ona jedynie źródłem wartości dla otoczenia, a więc jej metody nie pobierają żadnych argumentów owego typu. Dlatego gdy zadeklarujemy w klasie pole o typie kowariantnym, musi ono być „val”, czyli niezmiennie. Nie powstanie wówczas seter dla tego pola.

Komunikat oznacz, że w miejscu gdzie oczekiwano typu kontrawariantnego, znajduje się typ kowariantny. Ma to miejsce właśnie w seterze.

Dla

$S <: T$,

$\text{GenericCellMut}[S] <: \text{GenericCellMut}[T]$,

a więc:

$\text{GenericCellMut}[S].f(x:S): \text{Unit} <: \text{GenericCellMut}[T].f(x:T): \text{Unit}$,

co jednak nie może zajść, gdyż musiałaby zachodzić zależność:

$S >: T$ (kontrawariantność)

- b) Tak, da się poprawić:

```
class GenericCellMut[+T] (newX: T)
{
  private[this] var X: T = newX;
  def x: T = X
  def x_=[S >: T](newX: S): Unit = X = newX.asInstanceOf[T]
}
```

- c) Wersja kontrawariantna również się nie skompiluje, gdyż napotka analogiczny problem co kowariantna, ale nie w setterze, tylko w getterze. Parametr typu danej klasy nie może być kontrawariantny, jeżeli może ona być nie tylko odbiorcą wartości z otoczenia, ale jest też ich źródłem.

Dla
S>:T,
GenericCellMut[S]>: GenericCellMut[T],
a więc:
GenericCellMut[S].f():S >: GenericCellMut[T].f():T,
co jednak nie może zajść, gdyż musiałaby zachodzić zależność:
S<:T (kowariantność)

Zadanie 2

```
abstract class Sequence[+A] {  
  def append[B >: A](x: Sequence[B]): Sequence[A]  
}
```

Metoda append ma za zadanie zwrócić obiekt, co pasuje do kowariantności podanego typu "+A". Musi też pobrać argument, którego typ powinien być kontrawariantny, a taki nie jest. Aby rozwiązać ten problem, należy zastosować polimorfizm ograniczeniowy i przypisać typ również metodzie. Ograniczając typ w sposób [B>:A] pozwalamy na kowariantność tej metody.