

Imię i nazwisko	Data	Godzina
Mateusz Grzesiuk	17.11.2020r.	13:15

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

Lista 6

Zadanie 1

Funkcja `whileLoop` ma typ:

```
def whileLoop(f: => Boolean)(w: => Unit): Unit
```

Argumenty 'w' i 'f' mają takie, a nie inne typy, gdyż mają być funkcjami wywoływanymi przez nazwę. Nie chcemy też żeby argument był ewaluowany natychmiast, żebyśmy mogli do następnych, rekurencyjnych wywołań funkcji, przekazać jego samego, a nie jego wartość. Wartość zwracana przez 'whileLoop' musi być po prostu Unit, gdyż sama pętla niczego nie zwraca.

Zadanie 3

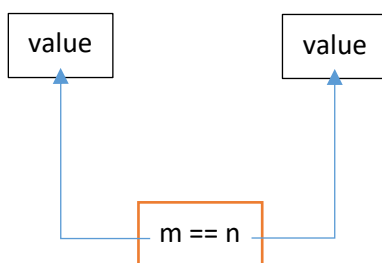
Program wydrukuje:

```
true
```

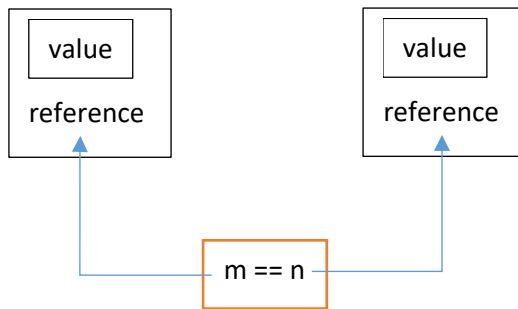
```
false
```

Porównując wyrażenia typu 'int', porównujemy ich wartości, ale porównując wyrażenia typu 'Integer' porównujemy ich wskaźniki na 2 obiekty.

Int:



Integer:



Zadanie 4

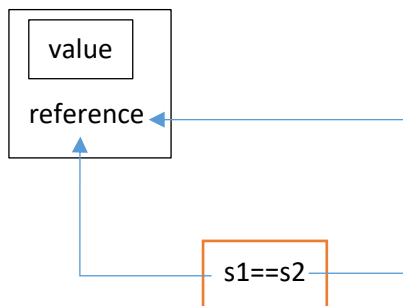
Program wydrukuje:

```
true
true
false
true
```

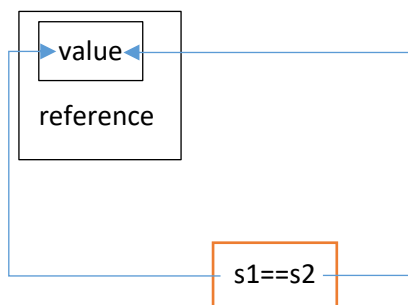
Kiedy tworzymy 2 obiekty typu 'String' (Bez użycia 'new') o tej samej zawartości, dokonywana jest optymalizacja i otrzymujemy referencję do tego samego obiektu. Jeżeli jednak użyjemy 'new' przy tworzeniu nowego 'String', mimo wszystko zostanie stworzony całkowicie nowy obiekt. Ze względu na to, porównując '==' obiekty stworzone bez 'new' uzyskamy wartość 'true', a gdy przynajmniej 1 stworzony używając 'new', otrzymamy 'false' – różne referencje.

Istnieje jednak metoda 'equals', która porównuje jedynie przechowywane wartości, ona w obu przypadkach zwróci 'true'.

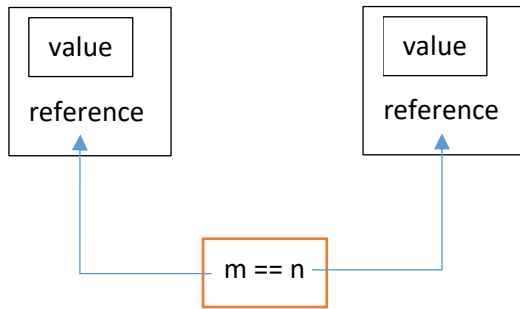
s1==s2 :



s1.equals(s2):



s1==s3:



s1.equals(s3):

