**Angelika Kiełbasa**

**Challenges 1**

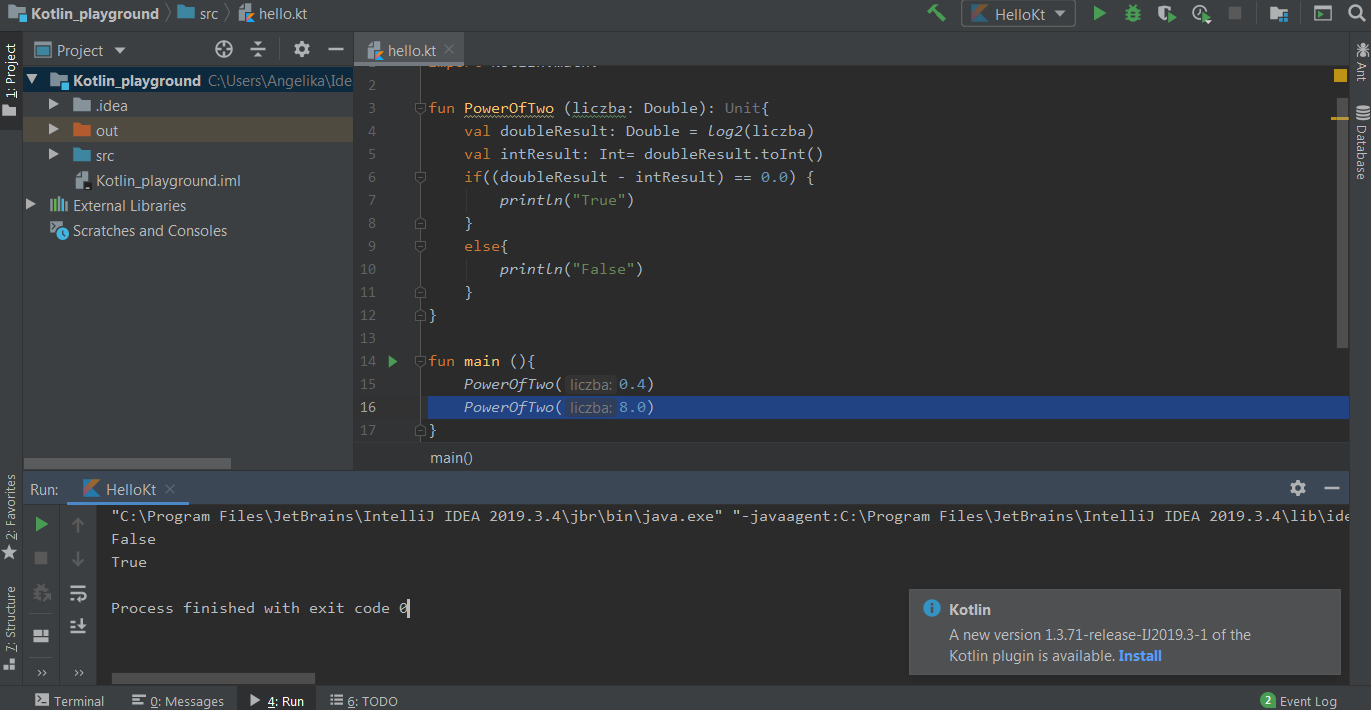
**Zadanie 1**

Po próbie wykonania danego kodu, wyświetla się: *"Error: Kotlin: Unresolved reference: lastName".* Jest to spowodowane tym, że do lastName nie została przypisana żadna wartość. Aby kod zadziałam poprawnie należałoby zadeklarować wartość var lastName i przypisać jej jakąś początkową wartość.

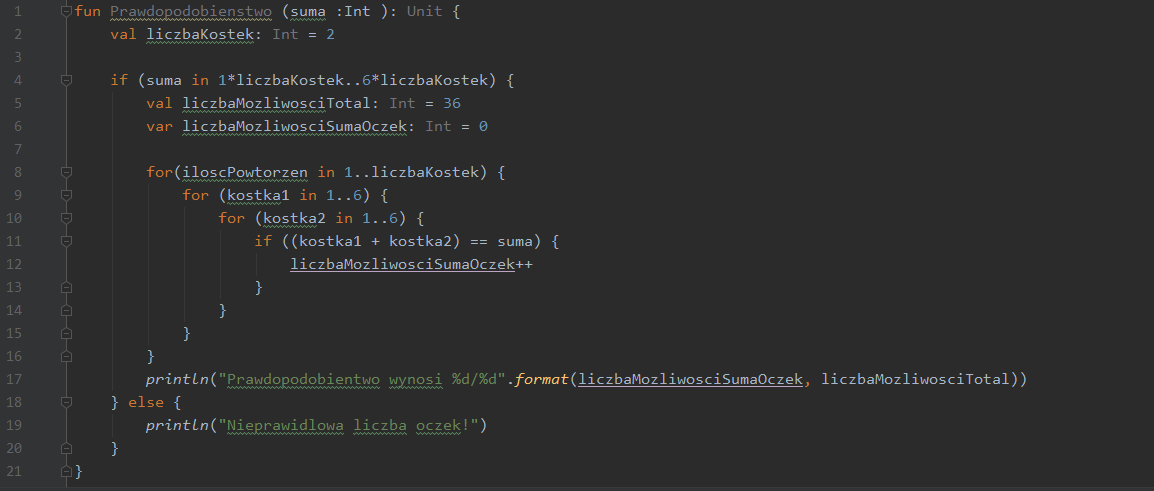
**Zadanie 2**

true  
false  
true  
true  
false

**Zadanie 3**



**Zadanie 4**



**Challenges 2**

**Zadanie 1**

sum = 55  
iterations = 11

**Zadanie 2**

Ranges

Jest to ciąg wartości między zdefiniowanymi dolnymi i górnymi granicami oraz skokiem. Mogą uprościć obliczenia, skrócić kod.

**Zadanie 3**

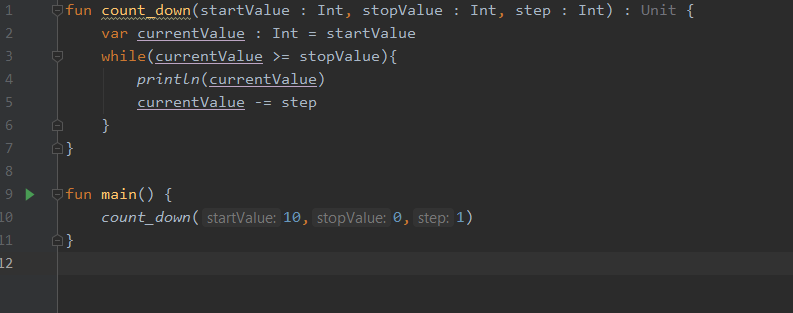
Ponieważ zawsze w skład danego zakresu będzie wchodziła przynajmniej jedna liczba, która może być jednocześnie dolną i górną granicą zakresu.

**Zadanie 4**

*When* wywołuje odpowiedni fragment kodu w zależności od podanej do niej wartości .

Funkcja When w Kotlinie różni się od funkcji Switch w C++ tym, że w funkcji Switch, w każdym przypadku (case) musi być zdeklarowana konkretna wartość, aby wywołać dalszą część kodu, natomiast w funkcji When wartość deklarujemy na wejściu, a zdeklarowane przypadki mogą obejmować na przykład całe zakresy w jakich mieści się dana liczba.

**Zadanie 5**



**Challenges 3**

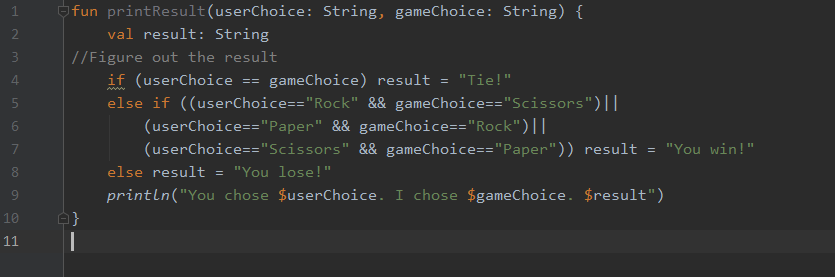
**Zadanie 1**

Funkcja doSomething zostanie skompilowana

Funkcja threeMultiplication nie zostanie skompilowana. Należy utworzyć nową wartość val x=x\*3.

Funkcja maxValue nie zostanie skompilowana. Należy dodać typ zwracanej wartości :Int

**Zadanie 2**



**Challenges 4**

**Zadanie 1**

Null Safety - Kompilator nie pozwoli przypisać null do referencji , jeżeli nie poinformujemy go o możliwości robienia tego. Należy "poinformować" go na etapie definicji referencji przy użyciu znaku ”?”.

Nullable Type - to typ upoważniony do przechowywania null

**Zadanie 2**

Elvison Operator to krótszy sposób zdefiniowania instrukcji warunkowej if/else. W zależności od tego czy zadeklarowany warunek jest spełniony zwracana jest odpowiednia wartość. Jest użytecznym narzędziem w przypadku, gdy chcemy mieć pewność, że pobierana przez nas wartość nie będzie null. Sprawdza wartość przypisaną do referencji i jeśli jest równa null to zastępuje 'nic' zadeklarowaną domyślną wartością.

**Zadanie 3**

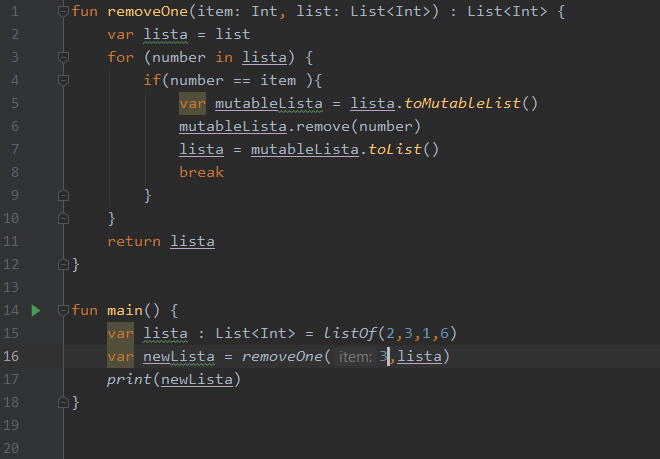
?. to bezpieczny operator wykorzystywany w celu sprawdzenia czy referencja przechowuje null.  
!! to mniej bezpieczny wariant . Działa na zasadzie: bez sprawdzania podaj wartość referencji, jeśli referencja to null pojawi się Error.

**Zadanie 4**

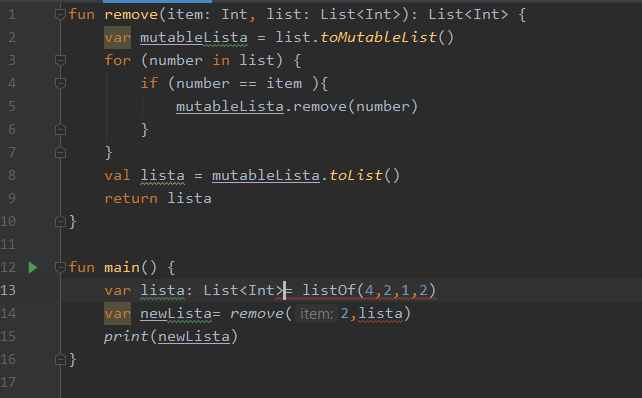
valid statements: name, distance , middleName

**Challenges 5**

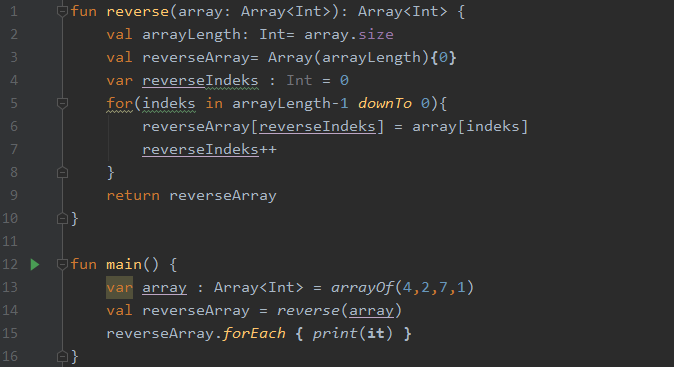
**Zadanie 1**



**Zadanie 2**



**Zadanie 3**



**Challenges 6**

**Zadanie 1**

Wyrażenie lambda to funkcje które nie posiadają nazwy, można je podawać jako wartości. Umożliwiają podanie całego bloku kodu np. jako argument metody.

**Zadanie 2**

"it" to skrót do każdego pojedynczego argumentu, który jest podawany do funkcji lambda. Pozwala na skrócony i wygodniejszy zapis w przypadku chęci dokonania tej samej operacji na wszystkich podanych argumentach

**Zadanie 3**

