

Wariant 461

Napisz statyczną metodę generyczną `contains`, która przyjmuje tablicę elementów typu generycznego `T` oraz jeden dodatkowy argument typu `T`. Metoda powinna zwracać `true`, jeśli dany element występuje w tablicy, w przeciwnym razie `false`. Wykorzystaj metodę `equals` do porównywania elementów. Stwórz przypadek testowy.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 462

Napisz statyczną metodę generyczną `printClass`, która przyjmuje jeden argument typu generycznego `T`. Metoda powinna wyświetlać nazwę klasy tego argumentu, wykorzystując metodę `getClass().getSimpleName()`. Stwórz przypadek testowy.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 463

Utwórz statyczną metodę generyczną `printArrayElements`, która przyjmuje tablicę elementów typu generycznego `T`. Metoda powinna wypisywać na ekranie wszystkie elementy tablicy, każdy w nowej linii. Jeśli tablica jest pusta, wyświetl komunikat "Pusta tablica". Stwórz przypadek testowy.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 464

Utwórz statyczną metodę generyczną `logElements`, która przyjmuje tablicę elementów typu generycznego `T`. Metoda powinna wypisywać każdy element tablicy na standardowe wyjście, zaczynając od "Element [indeks]: [element]". Na przykład, `logElements(new String[]{"apple", "banana"})` wypisze "Element 1: apple, Element 2: banana". Stwórz przypadek testowy.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 465

Utwórz statyczną metodę generyczną `areBothNull`, która przyjmuje dwa argumenty tego samego typu generycznego `T` i zwraca `true`, jeśli oba argumenty są `null`, a `false` w przeciwnym przypadku. Metoda ta ma oceniać, czy obie referencje są puste, bez porównywania ich faktycznych wartości. Na przykład, `areBothNull(object1, object2)` zwróci `true`, jeśli zarówno `object1` jak i `object2` są `null`. Stwórz przypadek testowy dla tej metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.