

მარტივი:

- **Generic Swap.** დაწერეთ Generic მეთოდი, რომელიც მიიღებს ორ ელემენტს და ნებისმიერი ტიპის მასივს, და ადგილებს გაუცვლის ამ ორ ელემენტს. დაწერეთ შესაბამისი ვალიდაციები.
- **Generic Maximum Finder.** დაწერეთ Generic მეთოდი, რომელიც მიიღებს ნებისმიერი ტიპის მასივს, რომელიც აიმპლემენტებს IComparable ინტერფეისს და მოძებნის მაქსიმალურ ელემენტს.

რთული:

- **Generic Queue.** ააწყვეთ generic queue რომელსაც ექნება enqueue, dequeue და peek ოპერაციები. Wiki: Queue არის FIFO (First In, First Out) ტიპის სტრუქტურა. Enqueue ოპერაციით ხდება Queue-ში ელემენტის დამატება, Peek-ით **მხოლოდ** ხდება ყველაზე ადრე დამატებული ელემენტის მიღება (მისი წაშლის გარეშე), ხოლო Dequeue-ს დროს ხდება ყველაზე ადრე დამატებული ელემენტის წაშლა და დაბრუნება.
- **Generic Stack.** ააწყვეთ generic stack, რომელსაც ექნება შემდეგი მეთოდები: Push, Pop, Peek. ამ მეთოდების შესახებ თქვენით მოიძიეთ ინფორმაცია.

ბონუსი (არ ფასდება)

- **Generic Binary Search Tree.** დაწერეთ generic binary search tree-ის იმპლმენტაცია, რომელსაც ექნება ნებისმიერი ტიპის ელემენტის (რომელიც აიმპლემენტებს IComparable ინტერფეისს) დამატების და მოძებნის ფუნქცია. ელემენტის წაშლის ფუნქციის დაწერა არ არის სავალდებულო.