

1. Record Nr.	UNINA9910975002303321
Autore	Mukherjee Sudipta
Titolo	NET 4.0 Generics : beginner's guide : enhance the type safety of your code and create applications easily using Generics in the .NET 4.0 Framework / / Sudipta Mukherjee
Pubbl/distr/stampa	Birmingham, U.K., : Packt Pub., 2012
ISBN	9786613453532 9781283453530 1283453533 9781849690799 1849690790
Edizione	[1st edition]
Descrizione fisica	1 online resource (396 p.)
Disciplina	005.2768
Soggetti	Generic programming (Computer science)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	"Learn by doing : less theory more results"--Cover. Includes index.
Nota di contenuto	Copyright; Credits; Foreword; About the Author; Acknowledgement; About the Reviewers; www.PacktPub.com; Table of Contents; Preface; Chapter 1: Why Generics?; An analogy; Reason 1: Generics can save you a lot of typing; Reason 2: Generics can save you type safety woes, big time; What's the problem with this approach?; Reason 3: Generics leads to faster code; Reason 4: Generics is now ubiquitous in the .NET ecosystem; Setting up the environment; Summary; Chapter 2: Lists; Why bother learning about generic lists?; Types of generic lists; Checking whether a sequence is a palindrome or not Time for action - creating the generic stack as the buffer Time for action - completing the rest of the method; Designing a generic anagram finder; Time for action - creating the method; Life is full of priorities, let's bring some order there; Time for action - creating the data structure for the prioritized shopping list; Time for action - let's add some gadgets to the list and see them; Time for action - let's strike off the gadgets with top-most priority after we have bought them; Time for action - let's create an appointment list; Live sorting and

statistics for online bidding

Time for action - let's create a custom class for live sorting
Why did we have three LinkedList as part of the data structure?; An attempt to answer questions asked by your boss; Time for action - associating products with live sorted bid amounts; Time for action - finding common values across different bidding amount lists; You will win every scrabble game from now on; Time for action - creating the method to find the character histogram of a word; Time for action - checking whether a word can be formed; Time for action - let's see whether it works

Trying to fix an appointment with a doctor? Time for action - creating a set of dates of the doctors' availability; Time for action - finding out when both doctors shall be present; Revisiting the anagram problem; Time for action - re-creating the anagram finder; Lists under the hood; Summary; Chapter 3: Dictionaries; Types of generic associative structures; Creating a tag cloud generator using dictionary; Time for action - creating the word histogram; Creating a bubble wrap popper game; Time for action - creating the game console; Look how easy it was!

How did we decide we need a dictionary and not a list? Let's build a generic autocomplete service; Time for action - creating a custom dictionary for autocomplete; Time for action - creating a class for autocomplete; The most common pitfall. Don't fall there!; Let's play some piano; Time for action - creating the keys of the piano; How are we recording the key strokes?; Time for action - switching on recording and playing recorded keystrokes; How it works?; C# Dictionaries can help detect cancer. Let's see how!; Time for action - creating the KNN API

Time for action - getting the patient records

Sommario/riassunto

This is a concise, practical guide that will help you learn Generics in .NET, with lots of real world and fun-to-build examples and clear explanations. It is packed with screenshots to aid your understanding of the process. This book is aimed at beginners in Generics. It assumes some working knowledge of C# , but it isn't mandatory. The following would get the most use out of the book: Newbie C# developers struggling with Generics. Experienced C++ and Java Programmers who are migrating to C# and looking for an alternative to other generic frameworks like STL and JCF would find this book handy.
