

Άσκηση 1: Βιβλιοθήκη Βιβλίων

Δημιουργήστε μια κλάση `Library` που θα περιέχει μια λίστα από βιβλία. Χρησιμοποιήστε έναν `Indexer` για να προσπελάζετε τα βιβλία με βάση τον αριθμό θέσης τους.

Λύση:

```
using System;
using System.Collections.Generic;

class Library
{
    private List<string> books = new List<string>();

    // Indexer για πρόσβαση σε βιβλία
    public string this[int index]
    {
        get
        {
            if (index < 0 || index >= books.Count)
            {
                throw new IndexOutOfRangeException("Λάθος δείκτης");
            }
            return books[index];
        }
        set
        {
            if (index < 0 || index >= books.Count)
            {
                throw new IndexOutOfRangeException("Λάθος δείκτης");
            }
            books[index] = value;
        }
    }

    // Μέθοδος για προσθήκη βιβλίου
    public void AddBook(string book)
    {
        books.Add(book);
    }
}

class Program
{
    static void Main()
    {
    }
```

```

Library library = new Library();
library.AddBook("Το Όνομα του Ρόδου");
library.AddBook("1984");
library.AddBook("Η Αλίκη στη Χώρα των Θαυμάτων");

// Χρήση του Indexer για πρόσβαση στα βιβλία
Console.WriteLine("Το πρώτο βιβλίο είναι: " + library[0]);
Console.WriteLine("Το δεύτερο βιβλίο είναι: " + library[1]);
Console.WriteLine("Το τρίτο βιβλίο είναι: " + library[2]);

// Αλλαγή του δεύτερου βιβλίου
library[1] = "Ο Μικρός Πρίγκιπας";
Console.WriteLine("Το νέο δεύτερο βιβλίο είναι: " + library[1]);
}
}

```

Άσκηση 2: Λεξικό Ορισμών

Δημιουργήστε μια κλάση `Dictionary` που θα περιέχει ορισμούς για διάφορες λέξεις. Χρησιμοποιήστε έναν `Indexer` για να προσπελάζετε τους ορισμούς με βάση τη λέξη.

Λύση:

```

using System;
using System.Collections.Generic;

class Dictionary
{
    private Dictionary<string, string> definitions = new Dictionary<string, string>();

    // Indexer για πρόσβαση στους ορισμούς
    public string this[string word]
    {
        get
        {
            if (definitions.ContainsKey(word))
            {
                return definitions[word];
            }
            return "Ορισμός δεν βρέθηκε";
        }
        set

```

```

{
    definitions[word] = value;
}
}
}

class Program
{
    static void Main()
    {
        Dictionary dictionary = new Dictionary();
        dictionary["Πολύμορφισμός"] = "Η ικανότητα ενός αντικειμένου να παίρνει πολλές μορφές";
        dictionary["Καψούρα"] = "Ερωτικό ενδιαφέρον για κάποιον ή κάτι";
        // Indexer για πρόσβαση στους ορισμούς
        Console.WriteLine("Πολύμορφισμός: " + dictionary["Πολύμορφισμός"]);
        Console.WriteLine("Καψούρα: " + dictionary["Καψούρα"]);
        Console.WriteLine("Συμβολισμός: " + dictionary["Συμβολισμός"]); // Δεν υπάρχει ορισμός για τη λέξη "Συμβολισμός"
    }
}

```

Άσκηση 3: Μητρώο Φοιτητών

Δημιουργήστε μια κλάση `StudentRegistry` που θα αποθηκεύει τα στοιχεία των φοιτητών (όνομα και βαθμολογία). Χρησιμοποιήστε έναν `Indexer` για να προσπελάζετε τις βαθμολογίες των φοιτητών με βάση το όνομα τους.

Λύση:

```

using System;
using System.Collections.Generic;

class StudentRegistry
{
    private Dictionary<string, double> students = new Dictionary<string, double>();

    // Indexer για πρόσβαση στις βαθμολογίες των φοιτητών
    public double this[string name]
    {
        get
        {

```

```

if (students.ContainsKey(name))
{
    return students[name];
}
throw new KeyNotFoundException("Ο φοιτητής δεν βρέθηκε");
}
set
{
    students[name] = value;
}
}

// Μέθοδος για προσθήκη φοιτητή
public void AddStudent(string name, double grade)
{
    students[name] = grade;
}
}

class Program
{
    static void Main()
    {
        StudentRegistry registry = new StudentRegistry();
        registry.AddStudent("Γιώργος", 8.5);
        registry.AddStudent("Μαρία", 9.0);
        registry.AddStudent("Κώστας", 7.5);

        // Χρήση του Indexer για πρόσβαση στις βαθμολογίες
        Console.WriteLine("Η βαθμολογία του Γιώργου είναι: " + registry["Γιώργος"]);
        Console.WriteLine("Η βαθμολογία της Μαρίας είναι: " + registry["Μαρία"]);
        Console.WriteLine("Η βαθμολογία του Κώστα είναι: " + registry["Κώστας"]);

        // Αλλαγή της βαθμολογίας του Κώστα
        registry["Κώστας"] = 8.0;
        Console.WriteLine("Η νέα βαθμολογία του Κώστα είναι: " +
        registry["Κώστας"]);
    }
}

```

Άσκηση 4: Διαχείριση Καλαθιού Αγορών

Δημιουργήστε μια κλάση `ShoppingCart` που θα περιέχει προϊόντα και τις ποσότητες τους. Χρησιμοποιήστε έναν `Indexer` για να προσπελάζετε τις ποσότητες των προϊόντων με βάση το

όνομα τους.

Λύση:

```
using System;
using System.Collections.Generic;

class ShoppingCart
{
    private Dictionary<string, int> cart = new Dictionary<string, int>();

    // Indexer για πρόσβαση στις ποσότητες των προϊόντων
    public int this[string productName]
    {
        get
        {
            if (cart.ContainsKey(productName))
            {
                return cart[productName];
            }
            return 0;
        }
        set
        {
            cart[productName] = value;
        }
    }

    // Μέθοδος για προσθήκη προϊόντος
    public void AddProduct(string productName, int quantity)
    {
        if (cart.ContainsKey(productName))
        {
            cart[productName] += quantity;
        }
        else
        {
            cart[productName] = quantity;
        }
    }
}

class Program
{
```

```

static void Main()
{
    ShoppingCart shoppingCart = new ShoppingCart();
    shoppingCart.AddProduct("Μήλα", 10);
    shoppingCart.AddProduct("Πορτοκάλια", 5);

    // Χρήση του Indexer για πρόσβαση στις ποσότητες των προϊόντων
    Console.WriteLine("Ποσότητα Μήλων: " + shoppingCart["Μήλα"]);
    Console.WriteLine("Ποσότητα Πορτοκαλιών: " +
shoppingCart["Πορτοκάλια"]);

    // Προσθήκη επιπλέον ποσοτήτων
    shoppingCart["Μήλα"] += 5;
    shoppingCart["Πορτοκάλια"] += 3;

    // Ενημερωμένες ποσότητες
    Console.WriteLine("Νέα ποσότητα Μήλων: " + shoppingCart["Μήλα"]);
    Console.WriteLine("Νέα ποσότητα Πορτοκαλιών: " +
shoppingCart["Πορτοκάλια"]);
}
}

```

Άσκηση 5: Λογαριασμοί Τραπεζών

Δημιουργήστε μια κλάση `BankAccount` που θα αποθηκεύει το υπόλοιπο των λογαριασμών τραπεζών με βάση τον αριθμό λογαριασμού. Χρησιμοποιήστε έναν `Indexer` για να προσπελάζετε το υπόλοιπο κάθε λογαριασμού.

Λύση:

```

using System;
using System.Collections.Generic;

class BankAccount
{
    private Dictionary<string, double> accounts = new Dictionary<string,
double>();

    // Indexer για πρόσβαση στα υπόλοιπα των λογαριασμών
    public double this[string accountNumber]
    {
        get
        {

```

```

        if (accounts.ContainsKey(accountNumber))
        {
            return accounts[accountNumber];
        }
        throw new KeyNotFoundException("Ο λογαριασμός δεν βρέθηκε");
    }
    set
    {
        accounts[accountNumber] = value;
    }
}

```

```

// Μέθοδος για προσθήκη ή ενημέρωση λογαριασμού
public void AddOrUpdateAccount(string accountNumber, double balance)
{
    accounts[accountNumber] = balance;
}

```

```

class Program
{
    static void Main()
    {
        BankAccount bank = new BankAccount();
        bank.AddOrUpdateAccount("12345", 1500.50);
        bank.AddOrUpdateAccount("67890", 2500.75);

        // Χρήση του Indexer για πρόσβαση στα υπόλοιπα των λογαριασμών
        Console.WriteLine("Υπόλοιπο λογαριασμού 12345: " + bank["12345"]);
        Console.WriteLine("Υπόλοιπο λογαριασμού 67890: " + bank["67890"]);

        // Ενημέρωση υπολοίπου λογαριασμού
        bank["12345"] += 500.00;
        Console.WriteLine("Νέο υπόλοιπο λογαριασμού 12345: " +
bank["12345"]);
    }
}

```