

Άσκηση 1: Δημιουργία κλάσης Book

Περιγραφή: Δημιούργησε μια κλάση Book που θα έχει τις εξής ιδιότητες:

- Title (Τίτλος βιβλίου)
- Author (Συγγραφέας)
- Pages (Αριθμός σελίδων)

Στη συνέχεια, δημιούργησε μερικά αντικείμενα της κλάσης Book και εμφάνισε τις πληροφορίες τους.

Λύση:

```
using System;
```

```
public class Book
{
    // Ιδιότητες της κλάσης Book
    public string Title { get; set; }
    public string Author { get; set; }
    public int Pages { get; set; }

    // Μέθοδος για την εμφάνιση πληροφοριών του βιβλίου
    public void ShowInfo()
    {
        Console.WriteLine($"Τίτλος: {Title}, Συγγραφέας: {Author}, Σελίδες: {Pages}");
    }
}

class Program
{
    static void Main()
    {
        // Δημιουργία αντικειμένων της κλάσης Book
        Book book1 = new Book { Title = "1984", Author = "George Orwell", Pages = 328 };
        Book book2 = new Book { Title = "To Kill a Mockingbird", Author = "Harper Lee", Pages =
281 };

        // Εμφάνιση πληροφοριών των βιβλίων
        book1.ShowInfo();
        book2.ShowInfo();
    }
}
```

Άσκηση 2: Δημιουργία κλάσης Student

Περιγραφή: Δημιούργησε μια κλάση Student που θα έχει τις εξής ιδιότητες:

- Name (Όνομα)
- Age (Ηλικία)
- Grade (Βαθμός)

Πρόσθεσε μια μέθοδο που θα επιστρέφει αν ο μαθητής έχει περάσει (βαθμός ≥ 50) ή όχι.

Λύση:

```
using System;
```

```
public class Student
{
    // Ιδιότητες της κλάσης Student
    public string Name { get; set; }
    public int Age { get; set; }
    public double Grade { get; set; }

    // Μέθοδος για να ελέγξει αν ο μαθητής πέρασε
    public bool HasPassed()
    {
        return Grade >= 50;
    }

    // Μέθοδος για εμφάνιση των πληροφοριών του μαθητή
    public void ShowInfo()
    {
        Console.WriteLine($"Όνομα: {Name}, Ηλικία: {Age}, Βαθμός: {Grade}, Πέρασε: {HasPassed()}");
    }
}

class Program
{
    static void Main()
    {
        // Δημιουργία αντικειμένων της κλάσης Student
        Student student1 = new Student { Name = "Maria", Age = 20, Grade = 85 };
        Student student2 = new Student { Name = "John", Age = 19, Grade = 45 };

        // Εμφάνιση πληροφοριών των μαθητών
        student1.ShowInfo();
        student2.ShowInfo();
    }
}
```

Άσκηση 3: Δημιουργία κλάσης Rectangle

Περιγραφή: Δημιούργησε μια κλάση Rectangle που θα έχει τις εξής ιδιότητες:

- Length (Μήκος)
- Width (Πλάτος)

Πρόσθεσε δύο μεθόδους:

1. Μια μέθοδο για να υπολογίζει την **περίμετρο** του ορθογωνίου.
2. Μια μέθοδο για να υπολογίζει την **επιφάνεια** του ορθογωνίου.

Λύση:

using System;

```
public class Rectangle
{
    // Ιδιότητες της κλάσης Rectangle
    public double Length { get; set; }
    public double Width { get; set; }

    // Μέθοδος για τον υπολογισμό της περιμέτρου
    public double CalculatePerimeter()
    {
        return 2 * (Length + Width);
    }

    // Μέθοδος για τον υπολογισμό της επιφάνειας
    public double CalculateArea()
    {
        return Length * Width;
    }

    // Εμφάνιση πληροφοριών του ορθογωνίου
    public void ShowInfo()
    {
        Console.WriteLine($"Μήκος: {Length}, Πλάτος: {Width}, Περίμετρος: {CalculatePerimeter()}, Επιφάνεια: {CalculateArea()}");
    }
}

class Program
{
    static void Main()
    {
        // Δημιουργία αντικειμένου της κλάσης Rectangle
        Rectangle rect = new Rectangle { Length = 5, Width = 3 };
    }
}
```

```
        // Εμφάνιση πληροφοριών του ορθογωνίου  
        rect.ShowInfo();  
    }  
}
```

Άσκηση 4: Δημιουργία κλάσης BankAccount

Περιγραφή: Δημιούργησε μια κλάση BankAccount που θα έχει τις εξής ιδιότητες:

- AccountNumber (Αριθμός Λογαριασμού)
- Balance (Υπόλοιπο)

Πρόσθεσε δύο μεθόδους:

1. Μια μέθοδο για κατάθεση χρημάτων.
2. Μια μέθοδο για ανάληψη χρημάτων (μόνο αν υπάρχει αρκετό υπόλοιπο).

Λύση:

```
using System;
```

```
public class BankAccount  
{  
    // Ιδιότητες της κλάσης BankAccount  
    public string AccountNumber { get; set; }  
    public double Balance { get; private set; }  
  
    // Μέθοδος για κατάθεση χρημάτων  
    public void Deposit(double amount)  
    {  
        if (amount > 0)  
        {  
            Balance += amount;  
            Console.WriteLine($"{amount} κατατέθηκαν στον λογαριασμό. Νέο υπόλοιπο:  
{Balance}");  
        }  
    }  
  
    // Μέθοδος για ανάληψη χρημάτων  
    public void Withdraw(double amount)  
    {  
        if (amount > 0 && amount <= Balance)  
        {  
            Balance -= amount;  
            Console.WriteLine($"{amount} αναλήφθηκαν από τον λογαριασμό. Νέο υπόλοιπο:  
{Balance}");  
        }  
    }  
}
```

```

        else
        {
            Console.WriteLine("Ανεπαρκές υπόλοιπο.");
        }
    }
}

class Program
{
    static void Main()
    {
        // Δημιουργία αντικειμένου της κλάσης BankAccount
        BankAccount account = new BankAccount { AccountNumber = "12345" };

        // Κατάθεση και ανάληψη χρημάτων
        account.Deposit(1000);
        account.Withdraw(500);
        account.Withdraw(700);
    }
}

```

Άσκηση 5: Δημιουργία κλάσης Person

Περιγραφή: Δημιούργησε μια κλάση Person που θα έχει τις εξής ιδιότητες:

- **FirstName** (Όνομα)
- **LastName** (Επίθετο)
- **Age** (Ηλικία)

Πρόσθεσε μια μέθοδο που θα επιστρέφει το **πλήρες όνομα** (όνομα και επίθετο μαζί) και μια άλλη μέθοδο που θα ελέγχει αν το άτομο είναι ενήλικο (ηλικία ≥ 18).

Λύση:

```

using System;

public class Person
{
    // Ιδιότητες της κλάσης Person
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public int Age { get; set; }

    // Μέθοδος για την επιστροφή του πλήρους ονόματος
    public string GetFullName()
    {
        return $"{FirstName} {LastName}";
    }
}

```

```

    }

    // Μέθοδος για τον έλεγχο αν το άτομο είναι ενήλικο
    public bool IsAdult()
    {
        return Age >= 18;
    }

    // Εμφάνιση πληροφοριών του ατόμου
    public void ShowInfo()
    {
        Console.WriteLine($"Πλήρες Όνομα: {GetFullName()}, Ηλικία: {Age}, Ενήλικος:
{IsAdult()}");
    }
}

class Program
{
    static void Main()
    {
        // Δημιουργία αντικειμένου της κλάσης Person
        Person person = new Person { FirstName = "John", LastName = "Doe", Age = 25 };

        // Εμφάνιση πληροφοριών του ατόμου
        person.ShowInfo();
    }
}

```