

Рефакторинг

Произведите рефакторинг для следующих фрагментов кода.

Фрагмент 1

```
class DataOrg
{
    public string name;
    public int age, score;

    int nameLen;

    public string[] getStmt()
    {
        if (name != null)
        {
            string[] row = new string[3];
            row[0] = name;
            row[1] = $"{age * 0.83}";
            row[2] = DateTime.Now;

            return row;
        }

        return null;
    }

    public int calcNamlen()
    {
        if (name == null)
        {
            return -1;
        }
        else
        {
            if (name.Length < 3)
                return 0;

            if (age < 18 || age > 65)
                return 0;

            if (score == -1)
                return 0;

            nameLen = name.Length * 4;

            return 0;
        }
    }

    public void SetValue(string name, string value)
    {
        if (name.Equals("age"))
```

```

    {
        age = value;
        return;
    }
    if (name.Equals("score"))
    {
        score = value;
        return;
    }
}
}

```

Фрагмент 2

```

public class gMethods
{
    public string Name;

    private int price;
    private int amount;
    private string platform;

    public void PrintPack()
    {
        this.PrintBanner();

        // Print details.
        Console.WriteLine("name: " + this.name);
        Console.WriteLine("amount: " + this.GetOutstanding());
        Console.WriteLine("price: " + this.price);
        Console.WriteLine("platform: " + platform);
    }

    float GetAmnt()
    {
        if ((platform.ToUpper().IndexOf("PC") > -1) &&
            (Name.ToUpper().IndexOf("XX") > -1) && amount > 0)
            return amount * 0.956;

        double temp = amount * price;
        Console.WriteLine(temp);

        temp = 0.8*amount * price;
        Console.WriteLine(temp);

        return -1;
    }
}

```

Фрагмент 3

```

class fighter
{
    private int iDamage;
    public string sName;
}

```

```

public int fighterHealth
{ get; set; }

public int fighterDamage
{ get; set; }

public int Weapon_Status
{ get; set; }

void logStatus(string name, int age, int health, int damage, int weaponStatus)
{
    Console.WriteLine($"name:{name}, age:{age}, health:{health}, damage:{damage}, weaponStatus:
{weaponStatus}");
}
public int GetDamage()
{
    // Weapon_Status * 5
    // Console.WriteLine($"Get Damage {iDamage}");
    return iDamage;
}
void atck()
{
    Console.WriteLine("Go Attack!");
    // TO DO: implement attack
}
public void Attack()
{
    try
    {
        atck();
    }
    catch(Exception e)
    {
        Console.WriteLine($"Go Attack Exception: {e}");
        throw e;
    }
}
}

```