All Inclusive zadatak

1. **Napravite klasu „Voće“ koja ima:**

* jedan public podatak (atribut) „boja“
* konstruktor sa parametrom, koji dodeljuje vrednost atributu
* apstraktan metod koji ne prima parametre i ne vraća ništa.
* funkciju koja vraća dužinu (broj slova) atributa.

1. **Napravite interfejs koji ima dva metoda:**

* prvi metod ne prima parametre i vraća podatak tipa string.
* drugi metod ne prima parametre i ne vraća ništa.

1. **Napravite novu klasu koja nasleđuje klasu „Voće“ i implementira interfejs koji ste napravili .** Ova klasa ima:

* dva atributa koja smatrate važnim za objekat tipa „Voće“. Ime i tip atributa određujete vi sami.
* kod za inicijalizaciju svih atributa putem konstruktora. Podrazumevane vrednosti (default vrednosti) određujete vi sami, jedino za atribut „boja“ treba postaviti da njegova vrednost bude podrazumevano „bela“.
* ToString metodu, koja vraća vrednosti atributa koje ste definisali.
* metod koji vraća vrednost prvog atributa ove klase.
* metod koji vraća vrednost drugog atributa ove klase.
* metode koje se moraju implementirati. Napišite programski kod za ove metode:
  + apstraktna metoda iz klase „Voće“ treba da ispiše broj vašeg indeksa i vrednost atributa „boja“
  + dve metode iz interfejsa:
    - prva metoda vraća vrednost atributa „boja“ u obrnutom redosledu.
    - druga metoda sabira ASCII vrednosti slova u atributu „boja“ i ispisuje njihovu sumu.

1. **U posebnoj klasi (glavni program) testirajte prethodno definisanu klasu :**

* kreirajte objekat br. 1 bez zadatih vrednosti za atribute (default vrednosti).
* ispisati objekat br. 1 koristeći ToString metodu
* kreirajte objekat br. 2 sa zadatim proizvoljnim vrednostima za atribute.
* ispisati objekat br. 2 koristeći ToString metodu.
* za objekat br.2 pozvati (testirati) i ostale tri metode koje imate u klasi.

1. **Iz klase definisane u zadatku br. 3 nasleđivanjem kreirajte klasu koja sadrži:**

* kod za inicijalizaciju svih atributa putem konstruktora.
* ToString metodu.
* prepravite funkciju koju je ova klasa dobila nasleđivanjem od klase definisane u zadatku br. 3, (misli se na metodu koja vraća obrnutim redosledom vrednost atributa „boja“), tako da svako slovo boje voća ispiše u posebnom redu.

1. **U posebnoj klasi (glavni program) testirajte klasu iz zadatka br. 5 :**

* kreirajte objekat bez zadatih vrednosti za atribute.
* ispisati objekat (ToString metoda)
* testirati prepravljenu funkciju (iz tačke br. 5).

Klasa Voce

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AllInclusive

{

abstract class Voce

{

public string boja;

public Voce(string b)

{

boja = b;

}

public abstract void IndexBoja();

public int Duzina()

{

return boja.Length;

}

}

}

Interfejs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AllInclusive

{

internal interface Interface1

{

string ReversBoja();

int AsciiZbir();

}

}

Klasa Vockica

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AllInclusive

{

internal class Vockica:Voce, Interface1

{

public string naziv;

public int tezina;

public Vockica(string boja, string ime, int t) : base(boja)

{

tezina = t;

naziv = ime;

}

public Vockica() : base("crvena")

{

naziv = "jabuka";

tezina = 300;

}

public override string ToString()

{

return "naziv: " + naziv + " boja: " + boja + " tezina: " + tezina;

}

public string GetNaziv()

{

return naziv;

}

public int GetTezina()

{

return tezina;

}

public override void IndexBoja()

{

Console.WriteLine("Broj indexa: 433/89");

Console.WriteLine("Boja voca: " + boja);

}

public string ReversBoja()

//public virtual string ReversBoja()

{

string b = boja;

char[] bojaRevers= b.ToCharArray();

Array.Reverse(bojaRevers);

b=new string(bojaRevers);

return b;

}

public int AsciiZbir()

{

string b = boja;

int suma = 0;

foreach(var c in b)

{

suma = suma + (int)c;

}

return suma;

}

}

}

Klasa SuperVoce

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AllInclusive

{

internal class SuperVoce:Vockica

{

public SuperVoce() :base("plava", "sljiva", 50)

{

}

public SuperVoce(string boja, string ime, int tezina):base(boja, ime, tezina)

{

}

public override string ToString()

{

return "Supervoce je " + boja + " boje" + " zove se " + naziv + " i tesko je " + tezina + " gr";

}

public new string ReversBoja()

{

string b = boja;

char[] bojaRevers = b.ToCharArray();

for(int i=bojaRevers.Length-1; i>=0; i--)

{

Console.WriteLine(bojaRevers[i]);

}

return "";

}

}

}

Testiranje

namespace AllInclusive

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Testiranje Vockice");

Vockica v1 = new Vockica();

Console.WriteLine(v1.ToString());

Vockica v2 = new Vockica("zelena", "grozdje", 100);

Console.WriteLine(v2.ToString());

v2.IndexBoja();

Console.WriteLine("boja unazad: "+v2.ReversBoja());

Console.WriteLine("ascii zbir: " + v2.AsciiZbir());

Console.WriteLine("Testiranje SuperVoca");

SuperVoce sv = new SuperVoce();

SuperVoce sv1 = new SuperVoce("zuta", "dunja", 200);

Console.WriteLine(sv.ToString());

Console.WriteLine(sv1.ToString());

Console.WriteLine("boja unazad za superVoce: ");

sv1.ReversBoja();

}

}

}