Уровень 1 номер 5

using System;

using System.Collections.Generic;

namespace Level1

{

abstract class Subject

{

protected List<Student> \_students = new List<Student>();

public void AddStudent(Student student)

{

\_students.Add(student);

}

public abstract void PrintStudents();

}

class ComputerScience : Subject

{

public override void PrintStudents()

{

Console.WriteLine("Студенты по информатике:");

foreach (var student in \_students)

{

student.Print();

}

}

}

class Mathematics : Subject

{

public override void PrintStudents()

{

Console.WriteLine("Студенты по математике:");

foreach (var student in \_students)

{

student.Print();

}

}

}

class Student

{

private string \_name;

private int \_mark;

private int \_missed;

public Student(string name, int mark, int missed)

{

\_name = name;

\_mark = mark;

\_missed = missed;

}

public void Print()

{

Console.WriteLine($"{\_name} - оценка: {\_mark}, пропущенных занятий: {\_missed}");

}

}

class Program

{

static void Main(string[] args)

{

var computerScience = new ComputerScience();

computerScience.AddStudent(new Student("Иванов", 5, 3));

computerScience.AddStudent(new Student("Петров", 2, 17));

var mathematics = new Mathematics();

mathematics.AddStudent(new Student("Сидоров", 3, 6));

mathematics.AddStudent(new Student("Александров", 2, 9));

computerScience.PrintStudents();

mathematics.PrintStudents();

}

}

}

Уровень 2 номер 1

using System;

namespace Level2

{

internal class Program

{

abstract class Person

{

protected string \_name;

public Person(string name)

{

\_name = name;

}

public string GetName()

{

return \_name;

}

}

class Human : Person

{

public Human(string name) : base(name) { }

}

class Student : Person

{

private int \_studentID;

private double[] \_results;

public Student(string name, double[] results, int studentID) : base(name)

{

\_results = results;

\_studentID = studentID;

}

public int GetStudentID()

{

return \_studentID;

}

public double[] GetResults()

{

return \_results;

}

public virtual double CalculateAverage()

{

return (\_results[0] + \_results[1] + \_results[2] + \_results[3]) / 4;

}

public void Print()

{

Console.WriteLine($"{GetName()}: {\_studentID} - {CalculateAverage()}");

}

}

static void Main(string[] args)

{

Student[] students = { new Student("Иванов", new double[] { 3, 5, 5, 4 }, 1234), new Student("Петров", new double[] { 3, 5, 4, 4 }, 2345),

new Student("Сидоров", new double[] { 5, 4, 5, 5 }, 3456), new Student("Ермаков", new double[] { 3, 3, 3, 4 }, 4567),

new Student("Романов", new double[] { 3, 4, 3, 3 }, 5678), new Student("Александров", new double[] { 3, 4, 5, 5 }, 6789)};

Sort(students);

Console.WriteLine("ФИО: № студ. билета - оценка");

for (int i = 0; i < students.Length; i++)

{

if (students[i].CalculateAverage() >= 4)

{

students[i].Print();

}

}

}

static void Sort(Student[] arr)

{

for (int i = 1; i < arr.Length; i++)

{

for (int j = 1; j < arr.Length; j++)

{

if (arr[j].CalculateAverage() > arr[j - 1].CalculateAverage())

{

Student temp = arr[j];

arr[j] = arr[j - 1];

arr[j - 1] = temp;

}

}

}

}

}

}

Уровень 3 номер 4

using System;

namespace Level3

{

abstract class Athlete

{

protected string \_name;

protected int \_result;

public Athlete(string name, int result)

{

\_name = name;

\_result = result;

}

public abstract void Print();

}

class SkierWoman : Athlete

{

public SkierWoman(string name, int result) : base(name, result) { }

public override void Print()

{

Console.WriteLine($"{\_name} (Woman): {\_result}");

}

}

class SkierMan : Athlete

{

public SkierMan(string name, int result) : base(name, result) { }

public override void Print()

{

Console.WriteLine($"{\_name} (Man): {\_result}");

}

}

internal class Program

{

static void Main(string[] args)

{

SkierWoman[] groupAWomen = { new SkierWoman("Иванова", 9), new SkierWoman("Петрова", 4), new SkierWoman("Сидорова", 7), new SkierWoman("Романова", 10) };

SkierMan[] groupAMen = { new SkierMan("Егоров", 6), new SkierMan("Бурунов", 5), new SkierMan("Грязнов", 8), new SkierMan("Глушаков", 9) };

Console.WriteLine("Таблица результатов для лыжниц-1:");

PrintResults(groupAWomen);

Console.WriteLine("\nТаблица результатов для лыжников-1:");

PrintResults(groupAMen);

Console.WriteLine("\nОбщая таблица результатов для лыжниц:");

PrintResults(groupAWomen);

Console.WriteLine("\nОбщая таблица результатов для лыжников:");

PrintResults(groupAMen);

Console.WriteLine("\nОбщая таблица результатов для всех участников:");

Athlete[] allParticipants = Merge(groupAWomen, groupAMen);

PrintResults(allParticipants);

}

static void PrintResults(Athlete[] arr)

{

foreach (var participant in arr)

{

participant.Print();

}

}

static Athlete[] Merge(SkierWoman[] arr1, SkierMan[] arr2)

{

Athlete[] result = new Athlete[arr1.Length + arr2.Length];

Array.Copy(arr1, result, arr1.Length);

Array.Copy(arr2, 0, result, arr1.Length, arr2.Length);

return result;

}

}

}