Самостоятельная работа от 12.02.2020

Тема: Определение классов

Задание 1

Создайте в Visual Studio проект консольного типа.

Определите в проекте ряд классов, перечисленных ниже в задании.

Требования:

* каждый класс определить в отдельном файле.
* каждый класс снабдить переопределенным методом ToString(), возвращающим строку с полным описанием данных экземпляра класса.   
  Смотрите пример: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/how-to-override-the-tostring-method>  
  *Загугли доп. инфу. если этого мало для понимания принципа реализации ToString()*
* определите в каждом классе специальный конструктор, принимающий значения каждого поля/свойства класса
* в методе Main() создайте по одному экземпляру каждого определённого класса. Используя конструктор задайте каждому полю/свойству значения (адекватные, читаемые)
* используя метод ToString() получите строковое представление каждого объекта. Выведите в консоль полученное строковое представление каждого объекта.

Перечень классов:

1. Студент.   
   Свойства: Имя, Курс, Половая принадлежность (доступно только для чтения, т.е. readonly. *Загугли, если не знаешь, что это такое*)
2. Служащий  
   Свойства: Имя, Профессия, Рабочий стаж
3. Цех  
   Свойства: Строковый шифр, количество служащих
4. Книга  
   Свойства: Название (Заголовок), Список фио авторов, Стоимость
5. Зачёт  
   Свойства: ФИО экзаменуемого, ФИО экзаменатора, Дата, Оценка
6. Адрес  
   Свойства: Почтовый индекс, Город, Улица, Дом, квартира
7. Товар  
   Свойства: Наименование, Количество, Стоимость, Срок годности
8. Учебная группа  
   Свойства: Шифр, Количество студентов, Год формирования, Специальность
9. Денежная купюра  
   Свойства: Серия и номер, Наминал, Наминал прописью
10. Компьютерная игра  
    Свойства: Название, Фирма разработчик, Год издания, Жанр

Листинг класса Program, описывающий решение задачи 1:

class Program

{

static void Main(string[] args)

{

Student student = new Student("James",2,"Male");

Console.WriteLine(student.ToString());

Employee employee = new Employee("Mike", "actor", 3);

Console.WriteLine(employee.ToString());

Workshop workshop = new Workshop("GIN", 25);

Console.WriteLine(workshop.ToString());

Book book = new Book("Time", new string[] {"Tom Soer", "Van Gog", }, 5);

Console.WriteLine(book.ToString());

Record record = new Record("Ivanov Ivan Ivanovich", "Vlamdirov Vladimir Vladimirovich", new DateTime(23,05,07), 4);

Console.WriteLine(record.ToString());

Address address = new Address(0534, "RaconCity", "Lomonosov", 3, 17);

Console.WriteLine(address.ToString());

Product product = new Product("Сыр", 545, 327, 7);

Console.WriteLine(product.ToString());

StudyGroup studyGroup = new StudyGroup("GIN", 25, 21, "Engineer");

Console.WriteLine(studyGroup.ToString());

Banknote banknote = new Banknote(359, 500, "Пятьсот");

Console.WriteLine(banknote.ToString());

ComputerGame computerGame = new ComputerGame("GameOFTrone", "Ubisoft", 2019, "РПГ");

Console.WriteLine(computerGame.ToString());

Console.ReadKey();

}

}

Листинг класса Студент, описывающий решение задачи 1:

class Student

{

private string Name

{get; set;}

private int Curse

{get; set;}

public readonly string Gender;

public Student(string name, int curse, string gender)

{

Name = name;

Curse = curse;

Gender = gender;

}

public override string ToString()

{

return ($" Student: {Name}, {Curse}, {Gender}");

}

}

Листинг класса Служащий, описывающий решение задачи 1:

class Employee

{

private string Name

{ get; set;}

private string Profession

{ get; set; }

private int WorkExperience

{ get; set; }

public Employee(string name, string profession, int workexperiencer)

{

Name = name;

Profession = profession;

WorkExperience = workexperiencer;

}

public override string ToString()

{

return ($" Employee: {Name}, {Profession}, {WorkExperience} ");

}

}

Листинг класса Цех, описывающий решение задачи 1:

class Workshop

{

private string StringCipher

{ get; set; }

private int NumberOfEmployees

{ get; set; }

public Workshop(string stringCipher, int numberOfEmployees)

{

StringCipher = stringCipher;

NumberOfEmployees = numberOfEmployees;

}

public override string ToString()

{

return ($" Workshop: {StringCipher}, {NumberOfEmployees}");

}

}

Листинг класса Книга, описывающий решение задачи 1:

class Book

{

private string Name

{ get; set; }

private string[] ListAuthorsNames

{ get; set; }

private int Cost

{ get; set; }

public Book(string name, string[] listAuthorsNames, int cost)

{

Name = name;

ListAuthorsNames = listAuthorsNames;

Cost = cost;

}

public override string ToString()

{

return ($" Book: {Name}, {ListAuthorsNames}, {Cost}");

}

}

Листинг класса Зачёт, описывающий решение задачи 1:

class Record

{

private string NameTestSubject

{ get; set; }

private string NameExaminer

{ get; set; }

private DateTime Date

{ get; set; }

private int Grade

{ get; set; }

public Record(string nameTestSubject, string nameExaminer, DateTime date, int grade)

{

NameTestSubject = nameTestSubject;

NameExaminer = nameExaminer;

Date = date;

Grade = grade;

}

public override string ToString()

{

return ($" Record: {NameTestSubject}, {NameExaminer}, {Date}, {Grade}");

}

}

Листинг класса Адрес, описывающий решение задачи 1:

class Address

{

private int Postcode

{ get; set; }

private string Town

{ get; set; }

private string Street

{ get; set; }

private int House

{ get; set; }

private int Apartment

{ get; set; }

public Address(int postcode, string town, string street, int house, int apartment)

{

Postcode = postcode;

Town = town;

Street = street;

House = house;

Apartment = apartment;

}

public override string ToString()

{

return ($" Address: {Postcode}, {Town}, {Street}, {House}, {Apartment}");

}

}

Листинг класса Товар, описывающий решение задачи 1:

class Product

{

private string Name

{ get; set; }

private int Number

{ get; set; }

private int Cost

{ get; set; }

private int ExpirationDate

{ get; set; }

public Product(string name, int number, int cost, int expirationDate)

{

Name = name;

Number = number;

Cost = cost;

ExpirationDate = expirationDate;

}

public override string ToString()

{

return ($" Product: {Name}, {Number}, {Cost}, {ExpirationDate}");

}

}

Листинг класса Учебная группа, описывающий решение задачи 1:

class StudyGroup

{

private string Cipher

{ get; set; }

private int NumberStudents

{ get; set; }

private int YearFormation

{ get; set; }

private string Specialty

{ get; set; }

public StudyGroup(string cipher, int numberStudents, int yearFormation, string specialty)

{

Cipher = cipher;

NumberStudents = numberStudents;

YearFormation = yearFormation;

Specialty = specialty;

}

public override string ToString()

{

return ($" StudyGroup: {Cipher}, {NumberStudents}, {YearFormation}, {Specialty}");

}

}

Листинг класса Денежная купюра, описывающий решение задачи 1:

class Banknote

{

private int SeriesAndNumber

{ get; set; }

private int FaceValue

{ get; set; }

private string ParValue

{ get; set; }

public Banknote(int seriesAndNumber, int faceValue, string parValue)

{

SeriesAndNumber = seriesAndNumber;

FaceValue = faceValue;

ParValue = parValue;

}

public override string ToString()

{

return ($" Banknote: {SeriesAndNumber}, {FaceValue}, {ParValue}");

}

}

Листинг класса Компьютерная игра, описывающий решение задачи 1:

class ComputerGame

{

private string Name

{ get; set; }

private string CompanyDeveloper

{ get; set; }

private int YearPublishing

{ get; set; }

private string Genre

{ get; set; }

public ComputerGame(string name, string companyDeveloper, int yearPublishing, string genre)

{

Name = name;

CompanyDeveloper = companyDeveloper;

YearPublishing = yearPublishing;

Genre = genre;

}

public override string ToString()

{

return ($" ComputerGame: {Name}, {CompanyDeveloper}, {YearPublishing}, {Genre}");

}

}

Скриншот с результатом работы программы:

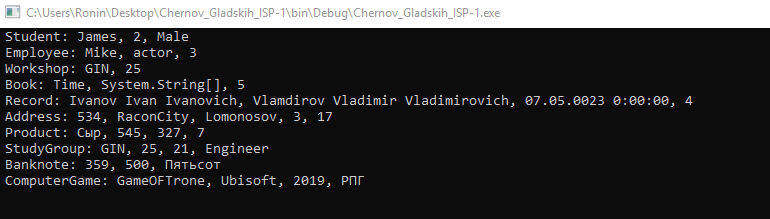


Рисунок 1 -результат работы программы