**Самостоятельная работа от 11.02.2020**

**Тема: Определение классов**

**Задание 1**

Создать в Visual Studio проект консольного типа.

Определите в проекте ряд классов:

1. Студент.   
   Свойства: Имя, Курс, Половая принадлежность (доступно только для чтения, т.е. readonly)

Класс Student:

class Student

{

public string Name { get; set; }

public int Course { get; set; }

readonly string Gender;

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

{

Name = name;

Course = course;

Gender = gender;

}

public override string ToString()

{

return $"Name: {Name} \nGender: {Gender} \nCourse: {Course}";

}

}

Main():

static void Main(string[] args)

{

Student student = new Student("Иван", 2, "Мужской");

Console.WriteLine(student);

Console.ReadKey();

}

Результат работы программы:

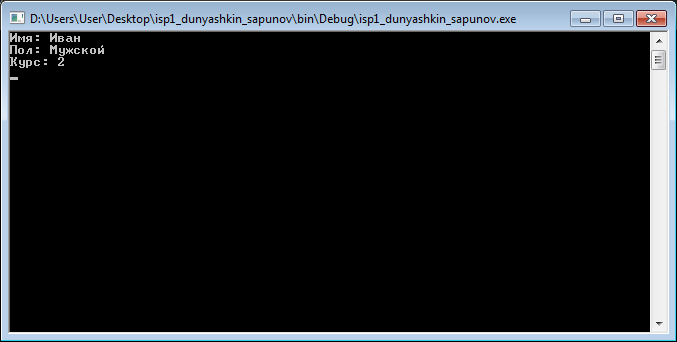


Рисунок 1 - результат описания класса Student

1. Служащий  
   Свойства: Имя, Профессия, Рабочий стаж

Класс Employee:

class Employee

{

public string Name { get; set; }

public string Profession { get; set; }

public int WorkingExperience { get; set; }

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

{

Name = name;

Profession = profession;

WorkingExperience = workingExperience;

}

public override string ToString()

{

return $"Имя: {Name} \nПрофессия: {Profession} \nРабочий стаж: {WorkingExperience}";

}

}

Main():

static void Main(string[] args)

{

Employee employee = new Employee("Андрей", "Уборщик", 7);

Console.WriteLine(employee);

Console.ReadKey();

}

Результат работы программы:

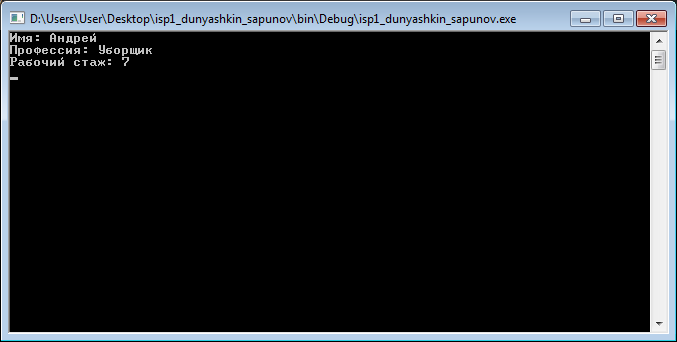


Рисунок 2 - результат описания класса Employee

1. Цех  
   Свойства: Строковый шифр, количество служащих

Класс Workshop:

class Workshop

{

public string StringCipher{ get; set; }

public int NumberOfEmployees { get; set; }

public Workshop(string stringCipher, int numberOfEmployees)

{

StringCipher = stringCipher;

NumberOfEmployees = numberOfEmployees;

}

public override string ToString()

{

return $"Строковый шифр: {StringCipher} \nКоличество служащих: {NumberOfEmployees}";

}

}

Main():

static void Main(string[] args)

{

Workshop workshop = new Workshop("193QP3", 45);

Console.WriteLine(workshop);

Console.ReadKey();

}

Результат работы программы:

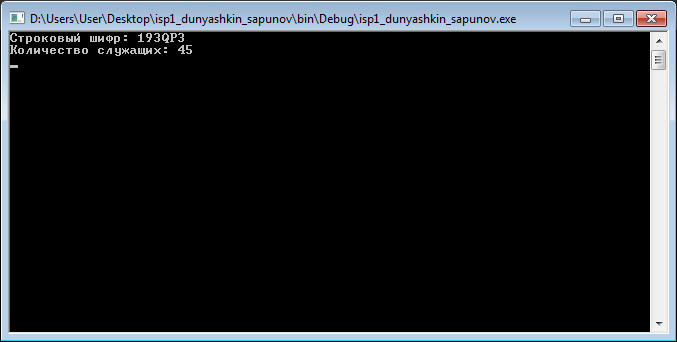


Рисунок 3 - результат описания класса Workshop

1. Книга  
   Свойства: Название (Заголовок), Список фио авторов, Стоимость

Класс Book:

class Book

{

public string Title { get; set; }

public string[] Authors { get; set; }

private string fioTemp;

public double Price { get; set; }

public Book (string title, double price, params string[] authors)

{

Title = title;

Authors = authors;

Price = price;

}

public override string ToString()

{

foreach (var fio in Authors)

{

fioTemp += fio + "; ";

}

return $"Title: {Title} \nPrice: {Price} \nAuthors: {fioTemp}";

}

}

Main():

static void Main(string[] args)

{

Book book = new Book("Колобок", 50.50, "Петр Петров Петрович",

"Игорь Игоревичев Игоревич");

Console.WriteLine(book);

Console.ReadKey();

}

Результат работы программы:

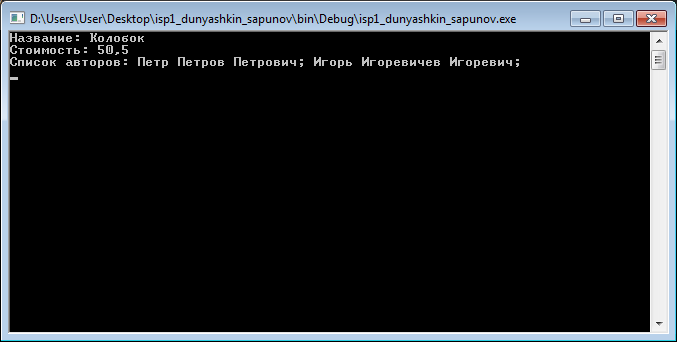


Рисунок 4 - результат описания класса Book

1. Зачёт  
   Свойства: ФИО экзаменуемого, ФИО экзаменатора, Дата, Оценка

Класс Test:

class Test

{

public string Examinator { get; set; }

public string Examiner { get; set; }

public DateTime Date { get; set; }

public int Estimation { get; set; }

public Test(string examinator, string examiner, DateTime date, int estimation)

{

Examinator = examinator;

Examiner = examiner;

Date = date;

Estimation = estimation;

}

public override string ToString()

{

return $"Examinator: {Examinator}, Examiner: {Examiner},

Date: {Date.ToLongDateString()}, Estimation: {Estimation}";

}

}

Main():

static void Main(string[] args)

{

Test test = new Test("Сергей Сергевьич Сергеев", "Петр Петров Петрович", new DateTime(2012, 5, 1), 5);

Console.WriteLine(test);

Console.ReadKey();

}

Результат работы программы:

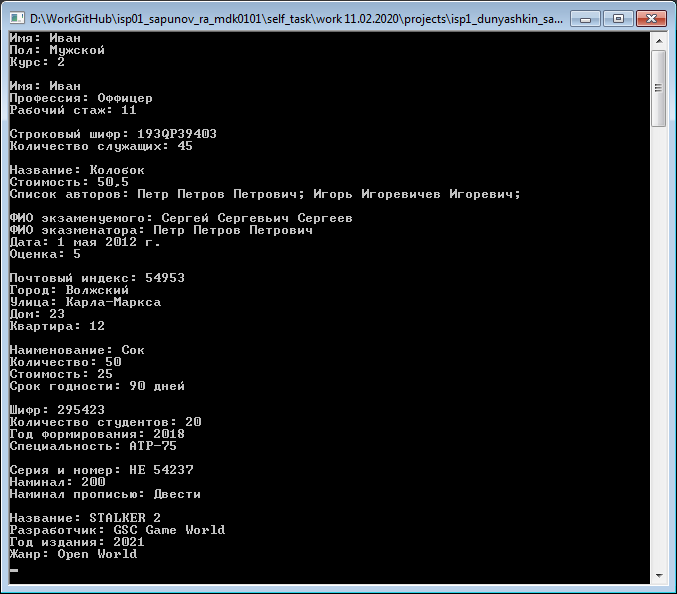


Рисунок 5 - результат описания класса Test

1. Адрес  
   Свойства: Почтовый индекс, Город, Улица, Дом, квартира

Класс Address:

class Address

{

public int Zip { get; set; }

public string City { get; set; }

public string Street { get; set; }

public int Home { get; set; }

public int ApartmentNumber { get; set; }

public Address(int zip, string city, string street, int home, int apartmentNumber)

{

Zip = zip;

City = city;

Street = street;

Home = home;

ApartmentNumber = apartmentNumber;

}

public override string ToString()

{

return $"Zip Code: {Zip}, City: {City}, Street: {Street}, Home: {Home}, ApartmentNumber: {ApartmentNumber}";

}

}

Main():

static void Main(string[] args)

{

Address address = new Address(54953, "Волжский", "Карла-Маркса", 23, 12);

Console.WriteLine(address);

Console.ReadKey();

}

Результат работы программы:

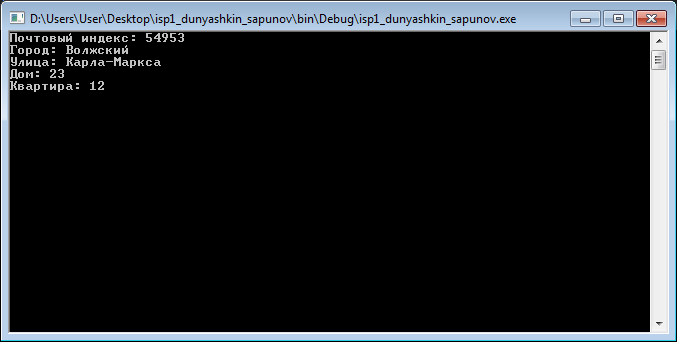


Рисунок 6 - результат описания класса Address

1. Товар  
   Свойства: Наименование, Количество, Стоимость, Срок годности

Класс Product:

class Product

{

public string Name { get; set; }

public int Quantity { get; set; }

public double Price { get; set; }

public int ExpirationDate { get; set; }

public Product(string name, int quantity, double price, int expirationDate)

{

Name = name;

Quantity = quantity;

Price = price;

ExpirationDate = expirationDate;

}

public override string ToString()

{

return $"Наименование: {Name} \nКоличество: {Quantity}

\nСтоимость: {Price} \nСрок годности: {ExpirationDate} дней";

}

}

Main():

static void Main(string[] args)

{

Product product = new Product("Сок", 50, 24.50, 90);

Console.WriteLine(product);

Console.ReadKey();

}

Результат работы программы:

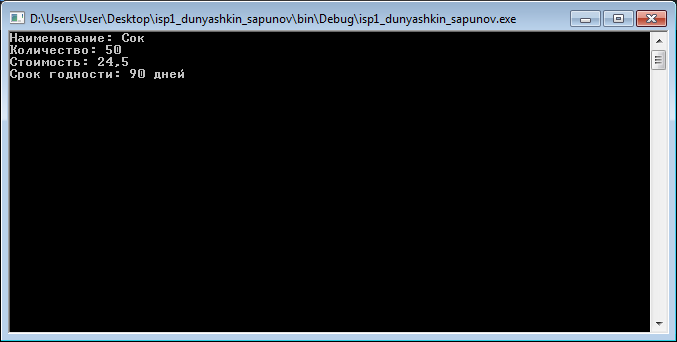


Рисунок 7 - результат описания класса Product

1. Учебная группа  
   Свойства: Шифр, Количество студентов, Год формирования, Специальность

Класс TrainingGroup:

class TrainingGroup

{

public string Сipher { get; set; }

public int NumberOfStudents { get; set; }

public int YearOfFormation { get; set; }

public string Specialty { get; set; }

public TrainingGroup(string cipher, int numberOfStudents,

int yearOfFormation, string specialty)

{

Сipher = cipher;

NumberOfStudents = numberOfStudents;

YearOfFormation = yearOfFormation;

Specialty = specialty;

}

public override string ToString()

{

return $"Шифр: {Сipher} \nКоличество студентов: {NumberOfStudents} \nГод формирования: {YearOfFormation} \nСпециальность: {Specialty}";

}

}

Main():

static void Main(string[] args)

{

TrainingGroup trainingGroup = new TrainingGroup("09.02.07", 10, 2018, "ИСП-1");

Console.WriteLine(trainingGroup);

Console.ReadKey();

}

Результат работы программы:

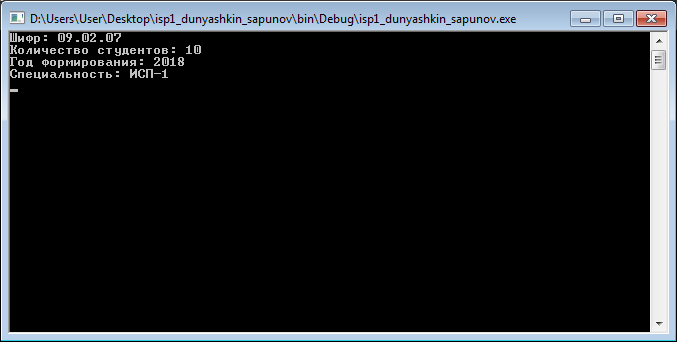


Рисунок 8 - результат описания класса TrainingGroup

1. Денежная купюра  
   Свойства: Серия и номер, Наминал, Наминал прописью

Класс MonetaryDenomination:

class MonetaryDenomination

{

public string SeriesAndNumber { get; set; }

public int Naminal { get; set; }

public string NaminalString { get; set; }

public MonetaryDenomination(string seriesAndNumber, int naminal,

string naminalString)

{

SeriesAndNumber = seriesAndNumber;

Naminal = naminal;

NaminalString = naminalString;

}

public override string ToString()

{

return $"Series And Number: {SeriesAndNumber}, Naminal: {Naminal}, Naminal String: {NaminalString}";

}

}

Main():

static void Main(string[] args)

{

MonetaryDenomination monetaryDenomination =

new MonetaryDenomination("HE 54237", 200, "Двести");

Console.WriteLine(monetaryDenomination);

Console.ReadKey();

}

Результат работы программы:

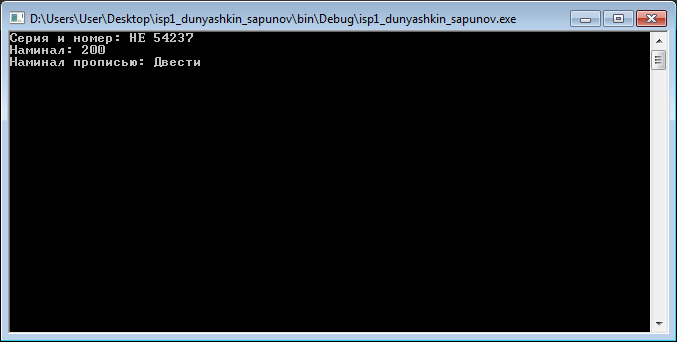


Рисунок 9 - результат описания класса MonetaryDenomination

1. Компьютерная игра  
   Свойства: Название, Фирма разработчик, Год издания, Жанр

Класс ComputerGame:

class ComputerGame

{

public string Name { get; set; }

public string DeveloperFirm{ get; set; }

public int PublicationDate { get; set; }

public string Genre { get; set; }

public ComputerGame(string name, string developerFirm, int publicationDate, string genre)

{

Name = name;

DeveloperFirm = developerFirm;

PublicationDate = publicationDate;

Genre = genre;

}

public override string ToString()

{

return $"Название: {Name} \nРазработчик: {DeveloperFirm}

\nГод издания: {PublicationDate} \nЖанр: {Genre}";

}

}

Main():

static void Main(string[] args)

{

ComputerGame computerGame = new ComputerGame("STALKER 2",

"GSC Game World", 2021, "Open World");

Console.WriteLine(computerGame);

Console.ReadKey();

}

Результат работы программы:

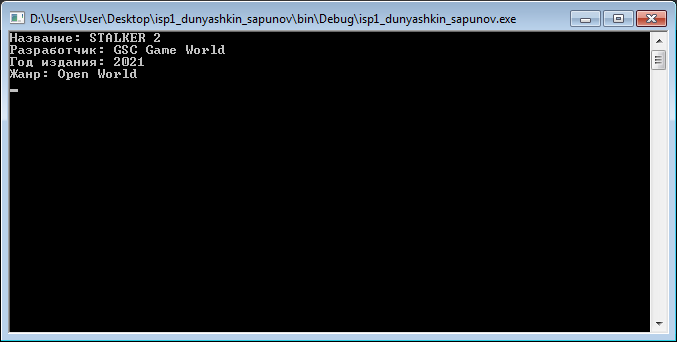


Рисунок 10 - результат описания класса ComputerGame