**Перечень задач для подготовки к промежуточной аттестации (экзамену)**

**по дисциплине «Основы алгоритмизации и программирования»**

**Задание 1.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс с именем Rectangle.

В теле класса создать два поля, описывающие длины сторон double side1, side2.

Создать пользовательский конструктор Rectangle(double side1, double side2), в теле которого поля side1 и side2 инициализируются значениями аргументов.

Создать два метода, вычисляющие площадь прямоугольника - double AreaCalculator() и периметр прямоугольника - double PerimeterCalculator().

Создать два свойства double Area и double Perimeter с одним методом доступа get.

Написать программу, которая принимает от пользователя длины двух сторон прямоугольника и выводит на экран периметр и площадь.

**Задание 2.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется спроектировать класс Book. Создать классы Title, Author и Content, каждый из которых должен содержать одно строковое поле и метод void Show().

Реализуйте возможность добавления в книгу названия книги, имени автора и содержания.

Выведите на экран разными цветами при помощи метода Show() название книги, имя автора и содержание.

**Задание 3.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать классы Point и Figure.

Класс Point должен содержать два целочисленных поля и одно строковое поле. Создать три свойства с одним методом доступа get. Создать пользовательский конструктор, в теле которого проинициализируйте поля значениями аргументов.

Класс Figure должен содержать конструкторы, которые принимают от 3-х до 5-ти аргументов типа Point. Создать два метода: double LengthSide (Point A, Point B), который рассчитывает длину стороны многоугольника; void PerimeterCalculator(), который рассчитывает периметр многоугольника.

Написать программу, которая выводит на экран название и периметр многоугольника.

**Задание 4.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс с именем Address.

В теле класса требуется создать поля: index, country, city, street, house, apartment.

Для каждого поля, создать свойство с двумя методами доступа.

Создать экземпляр класса Address.

В поля экземпляра записать информацию о почтовом адресе.

Выведите на экран значения полей, описывающих адрес.

**Задание 5.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс Converter.

В теле класса создать пользовательский конструктор, который принимает три вещественных аргумента, и инициализирует поля соответствующие курсу 3-х основных валют, по отношению к BYN – public Converter(double usd, double eur, double rub).

Написать программу, которая будет выполнять конвертацию из BYN в одну из указанных валют, также программа должна производить конвертацию из указанных валют в BYN.

**Задание 6.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс Employee.

В теле класса создать пользовательский конструктор, который принимает два строковых аргумента, и инициализирует поля, соответствующие фамилии и имени сотрудника.

Создать метод рассчитывающий оклад сотрудника (в зависимости от должности и стажа) и налоговый сбор.

Написать программу, которая выводит на экран информацию о сотруднике (фамилия, имя, должность), оклад и налоговый сбор.

**Задание 7.** Используя Visual Studio, создайте проект по шаблону Console Application

Требуется создать класс Invoice.

В теле класса создать три поля int account, string customer, string provider, которые должны быть проинициализированы один раз (при создании экземпляра данного класса) без возможности их дальнейшего изменения.

В теле класса создать два закрытых поля string article, int quantity

Создать метод расчета стоимости заказа с НДС и без НДС.

Написать программу, которая выводит на экран сумму оплаты заказанного товара с НДС или без НДС.

**Задание 8.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс User, содержащий информацию о пользователе (логин, имя, фамилия, возраст, дата заполнения анкеты).

Поле дата заполнения анкеты должно быть проинициализировано только один раз (при создании экземпляра данного класса) без возможности его дальнейшего изменения.

Реализуйте вывод на экран информации о пользователе.

**Задание 9.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс, представляющий учебный класс ClassRoom.

Создайте абстрактный класс ученик Pupil. В теле класса создайте свойство фамилии ученика, конструктор с параметром и абстрактные методы void Study(), void Read(), void Write(), void Relax().

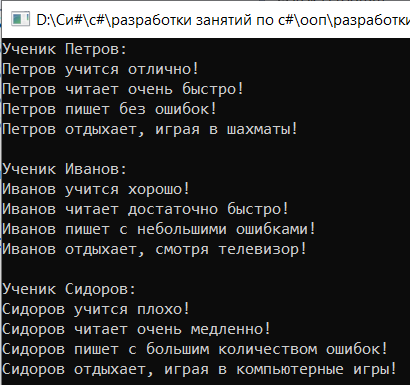
Создайте 3 производных класса ExcelentPupil, GoodPupil, BadPupil от класса базового класса Pupil и переопределите каждый из методов, в зависимости от успеваемости ученика.

Конструктор класса ClassRoom принимает аргументы типа Pupil, класс должен состоять из 4 учеников.

Предусмотрите возможность того, что пользователь может передать 2 или 3 аргумента.

Выведите информацию о том, как все ученики экземпляра класса ClassRoom умеют учиться, читать, писать, отдыхать.

Пример консольного вывода:



**Задание 10.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать абстрактный класс Vehicle.

В теле класса создайте поля: координаты и параметры средств передвижения (цена, скорость, год выпуска); конструктор с параметрами и метод вывода информации о транспортном средстве.

Создайте 3 производных класса Plane, Саг и Ship.

Для класса Plane должна быть определена высота и количество пассажиров.

Для класса Ship — количество пассажиров и порт приписки.

Написать программу, которая выводит на экран информацию о каждом средстве передвижения.

**Задание 11.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс DocumentWorker.

В теле класса создайте три метода OpenDocument(), EditDocument(), SaveDocument().

Тело каждого из методов добавьте вывод на экран соответствующих строк: "Документ открыт", "Редактирование документа доступно в версии Про", "Сохранение документа доступно в версии Про".

Создайте производный класс ProDocumentWorker.

Переопределите соответствующие методы, при переопределении методов выводите следующие строки: "Документ отредактирован", "Документ сохранен в старом формате, сохранение в остальных форматах доступно в версии Эксперт".

Создайте производный класс ExpertDocumentWorker от базового класса ProDocumentWorker.

Переопределите соответствующий метод. При вызове данного метода необходимо выводить на экран "Документ сохранен в новом формате".

В теле метода Main() реализуйте возможность приема от пользователя номера ключа доступа pro и exp.

Если пользователь не вводит ключ, он может пользоваться только бесплатной версией (создается экземпляр базового класса), если пользователь ввел номера ключа доступа pro и exp, то должен создаться экземпляр соответствующей версии класса, приведенный к базовому – DocumentWorker.

**Задание 12.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс Printer.

В теле класса создайте метод void Print(string value), который выводит на экран значение аргумента.

Реализуйте возможность того, чтобы в случае наследования от данного класса других классов, и вызове соответствующего метода их экземпляра, строки, переданные в качестве аргументов методов, выводились разными цветами.

Обязательно используйте приведение типов.

**Задание 13.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс AbstractHandler.

В теле класса создать методы void Open(), void Create(), void Chenge(), void Save().

Создать производные классы XMLHandler, TXTHandler, DOCHandler от базового класса AbstractHandler.

Написать программу, которая будет выполнять определение документа и для каждого формата должны быть методы открытия, создания, редактирования, сохранения определенного формата документа.

**Задание 14.** Используя Visual Studio, создайте проект по шаблону Console Application.

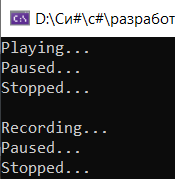
Требуется создать 2 интерфейса IPlayable (для проигрывания) и IRecodable (для записи).

В каждом из интерфейсов создайте по 3 метода void Play() / void Pause() / void Stop() и void Record() / void Pause() / void Stop() соответственно.

Создайте производный класс Player от базовых интерфейсов IPlayable и IRecodable.

Написать программу, которая выполняет проигрывание и запись.

Пример консольного вывода:



**Задание 15.** Используя Visual Studio, создайте проект по шаблону Console Application.

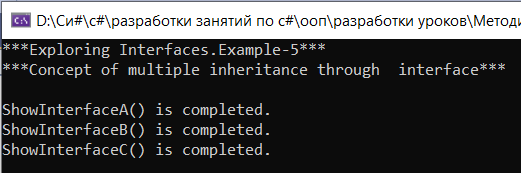
1. Создайте интерфейсы с именами InterfaceA, InterfaceB.

2. В каждом интерфейсе создайте соответствующие методы с именем ShowInterfaceA(), void ShowInterfaceB().

3. Создайте интерфейс с именем InterfaceC, реализующий множественное наследование.

4. Создайте класс MyClass, в котором реализуется интерфейс InterfaceС.

5. В методе Main() создайте объект myClassOb и вызовите все методы, чтобы в консольном окне был выведен следующий текст:

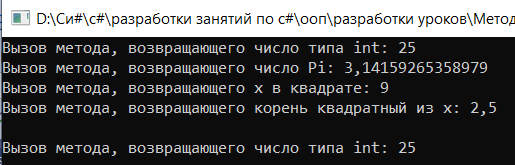


**Задание 16.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется объявить один интерфейс, в котором содержатся 4 метода:

1. метод, возвращающий число PI;
2. метод, возвращающий число типа int;
3. метод, возвращающий x в квадрате;
4. метод, возвращающий корень квадратный из x.

Реализовать интерфейс в классе MyClass, использующем этот интерфейс. Класс обязательно должен реализовать все методы интерфейса.



**Задание 17.** Используя Visual Studio, создайте проект по шаблону Console Application.

1. Создайте интерфейс ISwitchable, в котором объявите два метода – включение и выключение.
2. Реализуйте интерфейс в классе TVSet с методами включения и выключения телевизора.
3. Реализуйте интерфейс в классе PersonalComputer с методами включения и выключения компьютера.
4. С использованием списка вызовите методы созданных классов.

**Задание 18.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать класс Figura. Метод ShowArea перегружен. В зависимости от количества введенных параметров выводится название фигуры (один параметр – квадрат, два – прямоугольник, три – трапеция), вычисляется и выводится площадь.

**Задание 19.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется создать статический класс Calculator с методами для выполнения основных арифметических операций.

Написать программу, которая выводит на экран основные арифметические операции.

**Задание 20.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте структуру Book (поля: название, Ф.И.О. автора, год издания, количество страниц), содержащую конструктор и метод вывода информации об объекте. Создайте массив из *n* книг, выведите полную информацию обо всех книгах.

Пример консольного вывода:



**Задание 21.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте класс MyClass и структуру MyStruct, которые содержат в себе поля public string change.

В классе Program создайте два метода:

- static void ClassTaker(MyClass myClass), который присваивает полю change экземпляра myClass значение «изменено».

- static void StruktTaker(MyStruct myStruct), который присваивает полю change экземпляра myStruct значение «изменено».

Продемонстрируйте разницу в использовании классов и структур, создав в методе Main() экземпляры структуры и класса. Измените, значения полей экземпляров на «не изменено», а затем вызовите методы ClassTaker и StruktTaker. Выведите на экран значения полей экземпляров.

Проанализируйте полученные результаты.

**Задание 22.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте структуру с именем - Notebook.

Поля структуры: модель, производитель, цена.

В структуре должен быть реализован конструктор для инициализации полей и метод для вывода содержимого полей на экран.

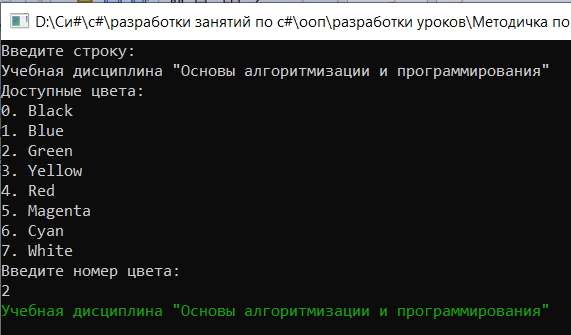
**Задание 23.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте статический класс с методом void Print (string stroka, int color), который выводит на экран строку заданным цветом.

Используя перечисление, создайте набор цветов, доступных пользователю.

Ввод строки и выбор цвета предоставьте пользователю.

Пример консольного вывода:

****

**Задание 24.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте перечисление, в котором будут содержаться должности сотрудников как имена констант.

Присвойте каждой константе значение, задающее количество часов, которые должен отработать сотрудник за месяц.

Создайте класс Accauntant с методом bool AskForBonus(Post worker, int hours), отражающий давать или нет сотруднику премию.

Если сотрудник отработал больше положенных часов в месяц, то ему положена премия.

Ввод должности сотрудника и количества отработанных часов предоставьте пользователю.

Пример консольного вывода:

|  |  |
| --- | --- |
|  |  |

**Задание 25.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется:

1. Описать класс «Клиент страховой компании».

*Поля класса*: имя клиента, вид страховки, размер страховки.

*Методы класса*: инициализация полей, получение информации о клиенте.

2. Создать массив клиентов страховой компании.

3. Получить ответы на следующие вопросы:

- кто из клиентов страховой компании застраховал автомобиль на сумму, превышающую 2000 руб.;

- сколько клиентов имеют автомобильную страховку.

**Задание 26.** Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется:

1. Описать класс «Туристическая фирма».

*Поля класса*: название, год создания, ИНН, прибыль за текущий год.

*Методы класса:* инициализация полей, получение информации о фирме.

2. Создать коллекцию объектов класса «Туристическая фирма», содержащую информацию о туристических фирмах города.

3. Для заданной фирмы найти ИНН и прибыль за текущий год.

4. Найти суммарную прибыль всех фирм.

5. Выяснить, какая из фирм получила наибольшую прибыль и каков размер этой прибыли.

6. Удалить из коллекции информацию об обанкротившейся фирме.

7. Добавить в коллекцию информацию о новой фирме.

**Задание 27.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте обобщенный список из названий цветов.

Выполните следующие манипуляции со списком с выводом каждого изменения содержимого списка:

1) добавьте в список элементы строчными буквами;

2) выведите содержимое коллекции на экран;

3) преобразуйте все буквы элементов в прописные;

4) выведите содержимое коллекции на экран;

5) определите текущее количество элементов коллекции;

6) добавьте в коллекцию новый элемент в конец списка;

7) выведите содержимое коллекции на экран;

8) добавьте в коллекцию новый элемент в начале списка соответствующего варианта;

9) выведите содержимое коллекции на экран;

10) удалите третий элемент коллекции;

11) выведите содержимое коллекции на экран;

12) расположите элементы в обратном порядке

**Задание 28.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создать класс Tour, в котором метод TourCalc перегружен. Стоимость тура вычисляется в зависимости от количества и типа введенных параметров: без параметров – Минское море, бесплатно; один параметр (страна) – 1 день, 50 руб; два параметра (страна, количество дней n) – стоимость = 50\*n. Продемонстрируйте работу объектов класса.

**Задание 29.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создается класс Tovar с полями: название name, цена price, количество kvo. Поля инкапсулируются с ограничениями (цена от 1 до 20, количество от 0 до 10). Вычисляется стоимость заказанного товара каждого вида и всего заказа. Продемонстрируйте работу объектов класса.

**Задание 30.** Используя Visual Studio, создайте проект по шаблону Console Application.

Спроектируйте класс Tovar с полями name (название), price (цена) и методом Calc (расчет и печать стоимости). Наследуйте от него классы Book (книги) c полем kvo (количество), Pen (ручки) c полем kvo и Сandy (конфеты) c полем ves (вес). Переопределите метод Calc для расчета стоимости каждого товара. Продемонстрируйте работу объектов класса.

**Задание 31.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте класс Calculator. В теле класса создайте четыре метода для арифметических действий: (Add – сложение, Sub – вычитание, Mul – умножение, Div – деление).

Метод деления должен делать проверку деления на ноль, если проверка не проходит, сгенерировать исключение.

Пользователь вводит значения, над которыми хочет произвести операцию и выбрать саму операцию.

При возникновении ошибок должны выбрасываться исключения.

**Задание 32.** Используя Visual Studio, создайте проект по шаблону Console Application.

Спроектируйте абстрактный класс Tovar и классы-потомки: Obuv (обувь), Odejda (одежда), Posuda (посуда). Задайте общие (название, цена) и особенные (количество, размер, масса, цвет) поля и свойства. В классе Tovar задайте абстрактный метод CalcSum (расчет стоимости) и обычный Print (вывод всей информации). Абстрактный метод CalcSum переопределяется для каждого товара. Продемонстрируйте работу объектов класса.

**Задание 33.** Используя Visual Studio, создайте проект по шаблону Console Application.

1. Создайте интерфейс с именем IInterface.
2. Создайте в интерфейсе метод Method().
3. Создайте производный класс DerivedClass, который реализует интерфейс IInterface.
4. Реализуйте метод Method() в классе DerivedClass, чтобы в консольном окне выводился текст:

Метод, реализованный в классе DerivedClass

1. В методе Main() создайте объект dClass и вызовите метод Method().

**Задание 34.** Используя Visual Studio, создайте проект по шаблону Console Application.

1. Создайте интерфейсы IInterface1, IInterface2, IInterface3.
2. Интерфейс IInterface2 наследуйте от интерфейса IInterface1.
3. Интерфейс IInterface3 наследуйте от интерфейса IInterface2.
4. В каждом интерфейсе создайте по одному методу с соответствующими именами: Method1(), Method2(), Method3().
5. Создайте производный класс DerivedClass, который наследуется от интерфейса IInterface3.
6. В классе DerivedClass реализуйте все три метода Method1(), Method2(), Method3().
7. В методе Main() создайте объект dClass и вызовите все методы, чтобы в консольном окне был выведен следующий текст:

Метод Method1

Метод Method2

Метод Method3

**Задание 35.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте класс Person со свойствами имени и возраста человека, реализующий интерфейс IComparable.

Создайте список и множество из четырех людей.

Отсортируйте объекты по имени в алфавитном порядке.

**Задание 36.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте абстрактный класс Человек, в котором будет свойство Имя и абстрактный метод СказатьПриветствие(), от этого класса будет несколько наследников, которые представляют национальность (русский, украинец, американец...). Должно получиться так, что при вызове метода СказатьПриветствие() выводилось приветствие на языке соответствующему национальности (Привет, Merhaba, Hi...).

**Задание 37.** Используя Visual Studio, создайте проект по шаблону Console Application.

1. Создать статический метод double  S (double  a, double b, double c), который возвращает площадь треугольника со сторонами a, b и c.
2. Объявить тип делегата, соответствующий этому методу.
3. Объявить сам делегат.
4. Применить в программе созданный делегат 3 раза.

**Задание 38.** Используя Visual Studio, создайте проект по шаблону Console Application.

Разработать класс для объекта Product: Наименование, Производитель, Цена, Срок хранения, Количество. Создать массив объектов. Вывести: а) список товаров для заданного наименования; б) список товаров для заданного наименования, цена которых не превышает указанной; в) список товаров, срок хранения которых больше заданного.

**Задание 39.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создается класс Computer с полями: модель model, объем ОЗУ ram и жесткого диска hdd. Поля инкапсулируются с ограничениями (объем ОЗУ от 2 до 32 Гбайт, жесткого диска от 200 до 2000 Гбайт). Вычисляется стоимость указанной модели и общая стоимость всех моделей.

**Задание 40.** Используя Visual Studio, создайте проект по шаблону Console Application.

1. Создать статический метод double Average (int m, int n), который вычисляет и возвращает среднее арифметическое всех целых чисел от *m* до *n* включительно.
2. Объявить тип делегата, соответствующий этому методу;
3. Объявить сам делегат.
4. Применить в программе созданный делегат 3 раза.

**Задание 41.** Используя Visual Studio, создайте проект по шаблону Console Application.

Реализовать обобщенный класс Point, который определяет точку на координатной плоскости. В классе реализовать:

* внутренние поля x, y;
* конструктор с 2 параметрами;
* свойства доступа к внутренним полям класса;
* метод, выводящий значения внутренних полей класса.

Написать программу, иллюстрирующую работу данного класса. В качестве универсальных параметров используйте типы int, uint и double.

**Задание 42.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте обобщенный класс Triangle, описывающий треугольник. Треугольник задан сторонами *a*, *b* и *c*. Класс должен содержать:

* поля,
* свойства с дополнительной логикой,
* конструктор,
* метод вывода информации о треугольнике,
* метод, возвращающий площадь треугольника.

Написать программу, иллюстрирующую работу данного класса. В качестве универсальных параметров используйте типы int, uint и double.

**Задание 43.** Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте обобщенный класс Segment, описывающий отрезок на координатной плоскости. Отрезок задан координатами точек (*x*1; *y*1), (*x*2; *y*2). Класс должен содержать:

* поля,
* свойства,
* конструктор,
* метод вывода информации об отрезке,
* метод, возвращающий длину отрезка.

Написать программу, иллюстрирующую работу данного класса. В качестве универсальных параметров используйте типы int, long и double.

**Задание 44.** Используя Visual Studio, создайте проект по шаблону Console Application.

Разработать программу с использованием делегатов. В программе следует:

1. создать статический метод int Sum (int m, int n), который вычисляет и возвращает сумму всех целых чисел от *m* до *n* включительно;
2. объявить тип делегата, соответствующий этому методу;
3. объявить сам делегат;
4. применить в программе созданный делегат 3 раза.

**Задание 45.** Используя Visual Studio, создайте проект по шаблону Console Application.

Описать интерфейс IShape для геометрических фигур. Интерфейс должен содержать методы Perimeter и Area, возвращающие периметр и площадь соответственно.

Описать классы Triangle (треугольник) и Disk (круг), реализующие этот интерфейс. Параметры фигур должны задаваться при создании экземпляра.

Написать метод, принимающий фигуру и выводящий на экран её название, параметры, периметр и площадь.

Написать программу, использующую этот метод.

**Задание 46.** Используя Visual Studio, создайте проект по шаблону Console Application.

Описать интерфейс ICalculation для преобразований чисел типа double. Он должен содержать метод Perform (выполнить преобразование), принимающий число и возвращающий результат преобразования.

Описать два класса, реализующих этот интерфейс: Add и Multiply. Первый класс увеличивает число на некоторую величину, а второй умножает на заданный коэффициент (величины определяются конструкторами).

Написать метод Calculate, принимающий число и два преобразования. Метод должен последовательно применять преобразования к числу и возвращать результат.

Написать программу, использующую этот метод.

**Задание 47.** Используя Visual Studio, создайте проект по шаблону Console Application.

Спроектировать класс Zakaz с полями fam (фамилия), size (количество мест в номере), comfort (комфортность), методом Show (показать заказ) и четырьмя конструкторами с разным числом и типами параметров. Модифицируйте программу, добавив возможность многократного ввода заказа с клавиатуры (фамилия, количество мест в номере, комфорт). Завершение ввода – символ Q.

**Задание 48.** Используя Visual Studio, создайте проект по шаблону Console Application.

1. Создайте класс Купюра.

2. Добавьте поля, например: название, номинал, количество.

3. Для каждого поля создайте свойство с методами доступа get и set.

4. Обеспечьте проверку правильности вводимых данных.

5. Добавьте метод, выводящий на экран значения всех полей класса и продемонстрируйте его работу.

6. Создайте три объекта данного класса. Задайте значения свойствам класса и выведите значения на экран.

7. Добавьте конструктор по умолчанию и конструктор, принимающий один или несколько параметров.

8. Создайте объекты данного класса с помощью различных конструкторов.

9. Добавьте в класс методы, например:

-   изменить количество купюр, увеличив или уменьшив его на указанное число;

-   определить, достаточна ли сумма для оплаты покупки указанной стоимости.

10. Обеспечьте возможность продемонстрировать правильность работы созданных методов.

**Задание 49.** Используя Visual Studio, создайте проект по шаблону Console Application.

Описать класс Point (*Точка*), описывающий точку на плоскости. Определить в нем:

1. конструктор с параметрами, принимающий координаты точки;
2. статические поля, определяющие:

– количество точек 1-й координатной четверти;

– количество точек 2-й координатной четверти;

– количество точек 3-й координатной четверти;

– количество точек 4-й координатной четверти;

– общее количество точек;

1. метод для вывода координат точки;
2. статический метод, выводящий количество точек, расположенных в каждой координатной четверти, и общее количество точек.

Написать программу, использующую этот класс и методы.

**Задание 50.** Используя Visual Studio, создайте проект по шаблону Console Application.

1. Создать класс DateClass для работы с датой и разработать следующие элементы класса:

a. поле DateTime data;

b. конструктор без параметров;

с. конструктор, позволяющий установить заданную дату (конструктор с тремя параметрами – год, месяц, день);

d. свойство, позволяющее установить или получить значение поле класса (доступно для чтения и записи);

e. метод, позволяющий вычислить дату предыдущего дня;

f. метод, позволяющий вычислить дату следующего дня.

2. Создать объект класса с использованием конструктора с параметрами, введенными с клавиатуры.