**Практическая работа № 24**

Статические члены класса

***Цель работы:*** Получение практических навыков написания программ с использованием массивов объектов и статических членов класса.

Реализация трех задач практической работы должны быть через методы и иметь дружелюбный интерфейс.

|  |  |  |
| --- | --- | --- |
| №  п/п | Ф.И.О. | НОМЕР ЗАДАНИЯ |
|  | Балицкая Анастасия | 1 |
|  | Вартик Максим | 2 |
|  | Вдовиченко Ярослав | 3 |
|  | Веряскин Иван | 4 |
|  | Гоцалюк Арзу | 5 |
|  | Градинар Илья | 1 |
|  | Гребенщиков Данил | 2 |
|  | Завтонев Роман | 3 |
|  | Захаров Станислав | 4 |
|  | Ильинский Никита | 5 |
|  | Каплин Егор | 1 |
|  | Коваленко Михаил | 2 |
|  | Колак Михаил | 3 |
|  | Левицкая Мария | 4 |
|  | Милюкова Екатерина | 5 |
|  | Никифорова Елизавета | 1 |
|  | Подолян Евгений | 2 |
|  | Попов Максим | 3 |
|  | Родичев Максим | 4 |
|  | Рознован Роман | 5 |
|  | Тизул Игорь | 1 |
|  | Тихончук София | 2 |
|  | Толоченко Павел | 3 |
|  | Фокша Кирилл | 4 |
|  | Чебан Евгений | 5 |
|  | Чебан Илья | 1 |
|  | Юраш Ренат | 2 |

## **Статические члены и модификатор static**

полей, методов, свойств классы и структуры могут иметь статические поля, методы, свойства. Статические поля, методы, свойства относятся ко всему классу/всей структуре и для обращения к подобным членам необязательно создавать экземпляр класса / структуры.

### Статические поля

Статические поля хранят состояние всего класса / структуры. Статическое поле определяется как и обычное, только перед типом поля указывается ключевое слово **static**. Например, рассмотрим класс Person, который представляет человека:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | Person bob = new(68);  bob.СheckAge();     // Уже на пенсии    Person tom = new(37);  tom.СheckAge();     // Сколько лет осталось до пенсии: 28    // получение статического поля  Console.WriteLine(Person.retirementAge); // 65  // изменение статического поля  Person.retirementAge = 67;    class Person  {      int age;      public static int retirementAge = 65;      public Person(int age)      {          this.age = age;      }      public void СheckAge()      {          if (age >= retirementAge)              Console.WriteLine("Уже на пенсии");          else              Console.WriteLine($"Сколько лет осталось до пенсии: {retirementAge - age}");      }  } |

В данном случае класс Person имеет два поля: age (хранит возраст человека) и retirementAge (хранит пенсионный возраст). Однако поле retirementAge является статическим. Оно относится не к конкретному человеку, а ко всем людям. (В данном случае для упрощения пренебрежем тем фактом, что в зависимости от пола и профессии пенсионный возраст может отличаться.) Таким образом, поле retirementAge относится не к отдельную объекту и хранит значение НЕ отдельного объекта класса Person, а относится ко всему классу Person и хранит общее значение для всего класса.

Причем в самом классе мы можем использовать это поле как и любые другие. Так, в методе СheckAge(), который поверяет пенсионный статус человека, для проверки используем это поле:

|  |  |
| --- | --- |
| 1 | if (age >= retirementAge) |

Но если мы хотим обратиться к этому полю вне своего класса, то мы можем обращаться к этому полю по имени класса:

|  |  |
| --- | --- |
| 1  2 | Console.WriteLine(Person.retirementAge);  Person.retirementAge = 67; |

На уровне памяти для статических полей будет создаваться участок в памяти, который будет общим для всех объектов класса.

При этом память для статических переменных выделяется даже в том случае, если не создано ни одного объекта этого класса.

### Статические свойства

Подобным образом мы можем создавать и использовать статические свойства:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | Person bob = new(68);  bob.СheckAge();    Console.WriteLine(Person.RetirementAge); // 65    class Person  {      int age;      static int retirementAge = 65;      public static int RetirementAge      {          get { return retirementAge; }          set { if (value > 1 && value < 100) retirementAge = value; }      }      public Person(int age)      {          this.age = age;      }      public void СheckAge()      {          if (age >= retirementAge)              Console.WriteLine("Уже на пенсии");          else              Console.WriteLine($"Сколько лет осталось до пенсии: {retirementAge - age}") ;      }  } |

В данном случае доступ к статической переменной retirementAge опосредуется с помощью статического свойства RetirementAge.

Таким образом, переменные и свойства, которые хранят состояние, общее для всех объектов класса / структуры, следует определять как статические.

Нередко статические поля и свойства применяются для хранения счетчиков. Например, мы хотим иметь счетчик, который позволял бы узнать, сколько объектов Person создано:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | var tom = new Person();  var bob = new Person();  var sam = new Person();  Console.WriteLine(Person.Counter);  // 3    class Person  {      static int counter = 0;      public static int Counter => counter;      public Person()      {          counter++;      }  } |

В данном случае в классе Person счетчик хранится в приватной переменной counter, значение которой увеличивается на единицу при создании объекта в конструкторе. А с помощью статического свойства Counter, которое доступно только для чтения, мы можем получить значение счетчика.

### Статические методы

Статические методы определяют общее для всех объектов поведение, которое не зависит от конкретного объекта. Для обращения к статическим методам также применяется имя класса / структуры:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | Person bob = new(68);  Person.CheckRetirementStatus(bob);    class Person  {      public int Age { get; set; }      static int retirementAge = 65;      public Person(int age) => Age = age;      public static void CheckRetirementStatus(Person person)      {          if (person.Age >= retirementAge)              Console.WriteLine("Уже на пенсии");          else              Console.WriteLine($"Сколько лет осталось до пенсии: {retirementAge - person.Age}") ;      }  } |

В данном случае в классе Person определен статический метод CheckRetirementStatus(), который в качестве параметра принимает объект Person и проверяет его пенсионный статус.

Следует учитывать, что статические методы могут обращаться только к статическим членам класса. Обращаться к нестатическим методам, полям, свойствам внутри статического метода мы не можем.

### Статический конструктор

Кроме обычных конструкторов у класса также могут быть статические конструкторы. Статические конструкторы имеют следующие отличительные черты:

* Статические конструкторы не должны иметь модификатор доступа и не принимают параметров
* Как и в статических методах, в статических конструкторах нельзя использовать ключевое слово this для ссылки на текущий объект класса и можно обращаться только к статическим членам класса
* Статические конструкторы нельзя вызвать в программе вручную. Они выполняются автоматически при самом первом создании объекта данного класса или при первом обращении к его статическим членам (если таковые имеются)

Статические конструкторы обычно используются для инициализации статических данных, либо же выполняют действия, которые требуется выполнить только один раз

Определим статический конструктор:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | Console.WriteLine(Person.RetirementAge);    class Person  {      static int retirementAge;      public static int RetirementAge => retirementAge;      static Person()      {          if (DateTime.Now.Year == 2022)              retirementAge = 65;          else              retirementAge = 67;      }  } |

В данном случае с помощью встроенной структуры **DateTime** получаем текущий год. Для этого используется свойство DateTime.Now.Year. если он равен 2022, устанавливаем один пенсионный возраст. При другом значении текущего года устанавливается другое значение пенсионного возраста.

### Статические классы

Статические классы объявляются с модификатором static и могут содержать только статические поля, свойства и методы. Например, определим класс, который выполняет ряд арифметических операций:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | Console.WriteLine(Operations.Add(5, 4));         // 9  Console.WriteLine(Operations.Subtract(5, 4));    // 1  Console.WriteLine(Operations.Multiply(5, 4));    // 20    static class Operations  {      public static int Add(int x, int y) => x + y;      public static int Subtract(int x, int y) => x - y;      public static int Multiply(int x, int y) => x \* y; |

Задания для самостоятельной работы

Описать класс, соответствующий заданию. Создать массив из *N* объектов данного класса. При подсчете объектов использовать статические поля. При необходимости написать методы, реализующие операции с экземплярами класса. Написать программу решения задачи.

1. Ввести информацию по *N* ученикам (поля: порядковый номер, Ф.И.О. и три оценки). Определить количество отличников, хорошистов, троечников. Вывести список каждых.
2. Ввести информацию по *N* параллелограммам (поля: *a*, *b* – стороны параллелограмма, α – угол между сторонами). Определить количество квадратов, прямоугольников и ромбов. Вывести список каждых.
3. Ввести информацию по *N* треугольникам (поля: стороны треугольника). Определить количество равносторонних, равнобедренных и разносторонних треугольников. Вывести список каждых.
4. Ввести информацию по *N* треугольникам (поля: три угла треугольника). Определить количество остроугольных, прямоугольных и тупоугольных треугольников. Вывести список каждых.
5. Ввести информацию по *N* школьникам (поля: Ф.И.О, пол, год рождения). Определить количество мальчиков и девочек. Вывести список каждых.

Массивы объектов

***Цель работы:*** Овладение приемами разработки простейших классов на языке С#. Получение практических навыков написания программ с использованием массива объектов.

|  |  |  |
| --- | --- | --- |
| №  п/п | Ф.И.О. | НОМЕР ВАРИАНТА |
|  | Балицкая Анастасия | 11,1 |
|  | Вартик Максим | 6,2 |
|  | Вдовиченко Ярослав | 10,3 |
|  | Веряскин Иван | 7,4 |
|  | Гоцалюк Арзу | 6,5 |
|  | Градинар Илья | 9,2 |
|  | Гребенщиков Данил | 1,8 |
|  | Завтонев Роман | 8,10 |
|  | Захаров Станислав | 3,11 |
|  | Ильинский Никита | 9,3 |
|  | Каплин Егор | 1,12 |
|  | Коваленко Михаил | 4,5 |
|  | Колак Михаил | 7,1 |
|  | Левицкая Мария | 6,12 |
|  | Милюкова Екатерина | 2,9 |
|  | Никифорова Елизавета | 5,10 |
|  | Подолян Евгений | 5,11 |
|  | Попов Максим | 4,3 |
|  | Родичев Максим | 3,7 |
|  | Рознован Роман | 8,9 |
|  | Тизул Игорь | 9,7 |
|  | Тихончук София | 4,12 |
|  | Толоченко Павел | 10,1 |
|  | Фокша Кирилл | 11,12 |
|  | Чебан Евгений | 5,2 |
|  | Чебан Илья | 2,3 |
|  | Юраш Ренат | 6,7 |

Задания для самостоятельной работы

**Описать класс, соответствующий заданию, содержащий поля, свойства, конструктор с параметрами. При необходимости вложить в свойства дополнительную логику проверки значений и написать методы, реализующие операции с экземплярами класса. Создать массив из *N* объектов данного класса. Написать программу решения задачи.**

1. Ввести информацию по *N* школьникам (Ф.И.О, пол, год рождения). Определить количество мальчиков и девочек. Вывести список каждых.

2. Ввести информацию по *N* студентам (№ группы, Ф.И.О. и три оценки). Вывести сведения о студентах, сдавших экзамены на 4 и 5.

3. Ввести информацию по *N* спортсменам (Ф.И.О., рост, вес). Вывести сведения о спортсменах, чей вес превышает 70 кг. Определить их количество.

4. Ввести информацию по *N* книгам (автор, название, издательство, год, количество страниц). Вывести список книг заданного автора.

5. Ввести информацию по *N* пациентам (фамилия, имя, отчество, номер медицинской карты, диагноз). Вывести список пациентов, имеющих заданный диагноз.

6. Ввести информацию по *N* перевозкам (№ рейса, пункт назначения, вес). Вывести сведения о перевозке с минимальным весом. Найти суммарный объем всех перевозок.

7. Ввести информацию по *N* фильмам (название, режиссер, год выпуска). Вывести сведения обо всех фильмах заданного года выпуска.

8. Ввести информацию о *N* рейсах самолетов (№ рейса, время вылета, время прилета, пункт назначения). Вывести информацию обо всех рейсах в заданный город.

9. Ввести информацию по *N* книгам (автор, название, издательство, год, количество страниц). Вывести список книг, изданных после заданного года.

10. Ввести информацию по *N* ученикам (порядковый номер, Ф.И.О. и три оценки). Определить количество отличников, хорошистов, троечников. Вывести список каждых.

11. Ввести информацию по *N* участникам соревнований (Ф.И.О., год рождения, результат). Вывести сведения о победителе (победителях) соревнований.

12. Ввести информацию о *N* предметах (порядковый номер, наименование, вес). Вывести информацию о предметах, чей вес больше среднего арифметического.