

#4 Перегрузка методов и операций. Вложенные типы.

Задание:

1. Спроектируйте класс (по вариантам, либо придумайте свой). Добавьте в класс необходимые свойства/поля. Определите «красивый» вывод информации об объектах вашего класса, используя переопределенный *ToString*.
2. Перегрузите для вашего класса операторы согласно вариантам. Дополнительно выберите один другой оператор и придумайте вариант его использования.
3. Создайте класс-коллекцию, который будет хранить объекты вашего класса и оперировать с ними. Определите:
 - a. Метод для циклического вывода информации обо всех объектах, содержащихся в коллекции;
 - b. Индексатор для доступа к элементам коллекции по индексу;
 - c. Методы добавления и удаления объектов в/из коллекцию(-и);
 - d. Осуществляйте проверку на валидность/допустимых принимаемых методами параметров.
4. Добавьте в класс-коллекцию вложенный класс *Information*, содержащий дату создания класса-коллекции и его владельца. Проинициализируйте данный класс.
5. Добавьте в класс-коллекцию переопределенный метод *ToString*, выводящий информацию о коллекции, используя класс *Information*.

Повышенный уровень:

1. Ознакомьтесь с типами **Func**, **Action** и **Predicate**
<https://docs.microsoft.com/en-us/dotnet/api/system.func-1?view=netframework-4.8>
<https://docs.microsoft.com/en-us/dotnet/api/system.action-1?view=netframework-4.8>
<https://docs.microsoft.com/en-us/dotnet/api/system.predicate-1?view=netframework-4.8>
2. Добавьте в класс-коллекцию, созданную выше, методы:
 - a. Поиска элементов, удовлетворяющих условию. Метод должен возвращать новую коллекцию, содержащую лишь те элементы старой, которые удовлетворяют некоему условию. Параметром должен быть предикат.
 - b. Преобразования элементов коллекции в другой тип. Метод должен возвращать новую коллекцию с преобразованными

элементами (можно использовать стандартные коллекции: Array, List, etc.). Параметром должна быть функция, принимающая объект вашего класса и возвращающая преобразованный тип в string.

- c. Метод *ForEach*. Метод должен принимать действие (action) и вызывать его для каждого элемента коллекции.

Вопросы:

1. Что такое вложенный класс? Перечислите свойства вложенных/внутренних свойств.
2. Какие виды операторов существуют? Назовите несколько примеров каждой группы.
3. Для чего используется перегрузка операторов?
4. Как используется ключевое слово **operator**?
5. Какие операторы можно перегрузить в C#? Какие операторы нельзя перегрузить?
6. Можем ли мы перегрузить оператор +=? Почему?
7. Можем ли мы перегрузить оператор **checked**? А **new**?
8. Можно ли перегрузить оператор **[]**? Если нет, то что является «обходным» способом его перегрузки?

Повышенный уровень:

1. В чем отличие между **Func**, **Action**, **Predicate**?
2. Продемонстрируйте создание объектов **Func**, **Action**, **Predicate**, а так же их использование: вызов, передача как параметров, использование лямбда-выражений.

Варианты заданий:

1. *Конус*; операторы:
 - a. ++ (высота)
 - b. * (int, радиус)
2. *Цилиндр*; операторы:
 - a. + (цилиндр, размер)
 - b. **int()** (высота)
3. *Куб*; операторы:
 - a. * (int, размер)
 - b. % (int, можно ли вписать в куб сферу заданного радиуса)
4. *Сфера*; операторы:
 - a. != (сфера, равенство)
 - b. -- (диаметр)
5. *Равнобедренный треугольник*; операторы:
 - a. < (равенство)
 - b. **bool()** (является ли треугольник египетским)
6. *Прямоугольный параллелепипед*; операторы:
 - a. >= (равенство)
 - b. **int()** (объем)
7. *Круг*; операторы:
 - a. ++ (радиус)
 - b. / (круг, разность площадей)
8. *Трапеция*; операторы:
 - a. != (трапеция, равенство)
 - b. + (int, высота)
9. *Квадрат*; операторы:
 - a. ++ (увеличение стороны)
 - b. % (int, уменьшение стороны)
10. *Ромб*; операторы:
 - a. * (int, увеличение сторон)
 - b. **int()** (площадь)
11. *Четырехугольная пирамида*; операторы:
 - a. ++ (сторона основания)
 - b. * (четырехугольная пирамида, размер)
12. *Прямоугольный треугольник*; операторы:
 - a. < (прямоугольный треугольник, неравенство)
 - b. **Int()** (радиус вписанной в треугольник окружности)

13. *Конус*; операторы:

- a. / (double, радиус),
- b. != (конус, равенство)

14. *Сфера*; операторы:

- a. * (int, объем)
- b. **int()** (объем)

15. *Трапеция*; операторы:

- a. **bool()** (можно ли вписать в трапецию окружность)
- b. **int()** (площадь)

16. *Цилиндр*; операторы:

- a. < (равенство)
- b. ++ (радиус основания)

17. *Круг*; операторы:

- a. ++ (радиус)
- b. * (круг, объем)

18. *Квадрат*; операторы:

- a. != (квадрат, равенство)
- b. + (int, сторона квадрата)

19. *Куб*; операторы:

- a. == (равенство),
- b. **int()** (объем)

20. *Сфера*; операторы:

- a. % (int, можно ли вписать сферу в куб со стороной, заданного размера)
- b. / (сфера, разность сфер)