

№ 4 Перегрузка операций, методы расширения и вложенные типы

Задание

- 1) Создать заданный в варианте класс. Определить в классе необходимые методы, конструкторы, индексаторы и заданные перегруженные операции. Написать программу тестирования, в которой проверяется использование перегруженных операций.
- 2) Добавьте в свой класс вложенный объект `Owner`, который содержит `Id`, имя и организацию создателя. Проинициализируйте его
- 3) Добавьте в свой класс вложенный класс `Date` (дата создания). Проинициализируйте
- 4) Создайте статический класс `StatisticOperation`, содержащий 3 метода для работы с вашим классом (по варианту п.1): сумма, разница между максимальным и минимальным, подсчет количества элементов.
- 5) Добавьте к классу `StatisticOperation` методы расширения для типа `string` и вашего типа из задания №1. См. задание по вариантам.

Вариант 1	<p>Класс – Одномерный массив. Дополнительно перегрузить следующие операции: <code>*</code> – умножение массивов; <code>true</code> – истина если массив не содержит отрицательных элементов, <code>int()</code> – операция приведения – возвращает размер массива; <code>==</code> – проверка на равенство; <code><</code> – сравнение.</p> <p>Методы расширения:</p> <ol style="list-style-type: none">1) Проверка на содержание определённого символа в строке2) Удаление отрицательных элементов
Вариант 2	<p>Класс – Одномерный массив. Дополнительно перегрузить следующие операции: <code>-</code> – разность со скалярным значением; <code>></code> – проверка на вхождение элемента; <code>!=</code> – проверка на неравенство массивов; <code>+</code> – объединение массивов</p> <p>Методы расширения:</p> <ol style="list-style-type: none">1) Удаление гласных из строки2) Удаление первых пяти элементов
Вариант 3	<p>Класс – множество Set. Дополнительно перегрузить следующие операции: <code>+</code> – добавить элемент в множество</p>

	<p>(типа set+item); + – объединение множеств; * – пересечение множеств; явный int()– мощность множества; false – проверка на принадлежность размера массива определенному диапазону</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Добавление занятой после каждого слова 2) Удаление повторяющихся из м множества
Вариант 4	<p>Класс – множество Set. Дополнительно перегрузить следующие операции: - – удалить элемент из множества (типа set-item); * – пересечение множеств; < – сравнение множеств; > – проверка на подмножество; &– придумайте использование.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Добавление точки в конце строки 2) Удаление нулевых элементов из множества
Вариант 5	<p>Класс – множество Set. Дополнительно перегрузить следующие операции: ->>– удалить элемент из множества (типа set-item); > – проверка на подмножество; != – проверка множеств на неравенство; << – добавить элемент в множество (типа set+item); %– пересечение множеств.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Поиск самого короткого слова 2) Упорядочивание множества
Вариант 6	<p>Класс – однонаправленный список List. Дополнительно перегрузить следующие операции: >> – удалить элемент в заданной позиции + – добавить элемент в заданную позицию, например, != – проверка на неравенство множеств.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Поиск самого длинного слова 2) Удаление последнего элемента из списка
Вариант 7	<p>Класс – множество Set. Дополнительно перегрузить следующие операции: ++ -добавление случайного элемента к множеству, + – объединение множеств; <= – сравнение множеств; неявный int()– мощность множества; % - доступ к элементу в заданной позиции.</p>

	<p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Шифрование строки 2) Проверка на упорядоченность множества
Вариант 8	<p>Класс – множество Set. Дополнительно перегрузить следующие операции: > – проверка на принадлежность (типа операции in множественного типа)* – пересечение множеств; < – проверка на подмножество; явный Date()– приведение типа</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Выделение первого числа, содержащегося в строке 2) Удаление положительных элементов из множества
Вариант 9	<p>Класс – однонаправленный список List. Дополнительно перегрузить следующие операции: + – объединить два списка; -- – удалить элемент из начала (--list); == – проверка на равенство; true – проверка, пустой ли список.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Выделение последнего числа, содержащегося в строке 2) Удаление заданного элемента из списка
Вариант 10	<p>Класс – список List. Дополнительно перегрузить следующие операции: + – добавить элемент в начало (item+list); -- – удалить первый элемент из списка (--list); != – проверка на неравенство; * - объединение двух списков.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Подсчет количества слов с заглавной буквы 2) Проверка на повторяющиеся элементы в списке
Вариант 11	<p>Класс – список List. Дополнительно перегрузить следующие операции: + – добавить элемент в конец (list+item); -- – удалить элемент из конца (типа list--); != – проверка на неравенство; true – проверка упорядоченности элементов.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Подсчет количества слов. 2) Проверка на нулевые элементы в списке
Вариант 12	<p>Класс - однонаправленный список List. Дополнительно</p>

	<p>перегрузить следующие операции: ! – инверсия элементов; + - объединить два списка; == - проверка на равенство; < - добавление одного списка к другому.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Усечение строки до заданной длины 2) Сумма элементов списка
Вариант 13	<p>Класс - стек Stack. Дополнительно перегрузить следующие операции: + - добавить элемент в стек; -- - извлечь элемент из стека; true - проверка, пустой ли стек; > - копирование одного стека в другой с сортировкой в возрастающем порядке.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Подсчет количества предложений 2) Определение среднего элемента стека
Вариант 14	<p>Класс - очередь Queue. Дополнительно перегрузить следующие операции: + - добавить элемент; -- - извлечь элемент; true - проверка, пустая ли очередь; < - копирование одной очереди в другую с сортировкой в убывающем порядке; неявный int() – мощность.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Индекс первой точки 2) Последний элемент очереди
Вариант 15	<p>Класс - Вектор. Дополнительно перегрузить следующие операции: + - сложение векторов; > - сравнение векторов; == - копирование вектора, true - проверка, пустой ли вектор</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Усечение строки с начала 2) Удаление положительных элементов из вектора
Вариант 16	<p>Класс - Матрица. Дополнительно перегрузить следующие операции: + - сложение матриц; - - удаление заданной строки; > - сравнение матриц по модулю; * - копирование матрицы.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Поиск номера автомобиля в строке (белорусского) 2) Сумма главной диагонали

Вариант 17	<p>Класс - Марица. Дополнительно перегрузить следующие операции: - - вычитания числа из всех элементов матрицы; ++ инкремент всех элементов матрицы; != - сравнение матриц по модулю; неявный int() – количество нулевых элементов в матрице.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Сумма целых чисел в строке 2) Проверка на квадратность матрицы
Вариант 18	<p>Класс - Марица. Дополнительно перегрузить следующие операции: + - сложение матриц; -- обнуление всех элементов матрицы; == - сравнение матриц по нулевому столбцу; явный int() – количество отрицательных элементов в матрице.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Выделение первого числа, содержащегося в строке 2) Обнуление отрицательных элементов матрицы
Вариант 19	<p>Класс - Марица. Дополнительно перегрузить следующие операции: < - сравнения матриц; -- приведение матрицы к единичному виду; == - сравнение матриц по первому элементу; * – инверсия всех элементов матрицы.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Разность двух первых целых чисел строки 2) Сумма элементов матрицы
Вариант 20	<p>Класс - стек Stack. Дополнительно перегрузить следующие операции: * - добавить элемент в стек; /- извлечь элемент из стека; true - проверка, есть ли в стеке отрицательные элементы; == - сравнения стеков.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Подсчет количества вопросительных предложений в строке 2) Проверка на ноль первого элемента
Вариант 21	<p>Класс - стек Stack. Дополнительно перегрузить следующие операции: - - извлечение всех элементов равных заданному; ++ - дублирование верхнего элемента; < копирование неповторяющихся элементов из второго стека.</p> <p>Методы расширения:</p>

	<ol style="list-style-type: none"> 1) Подсчет количества восклицательных предложений в строке 2) Проверка стека на наличие отрицательных элементов
Вариант 22	<p>Класс - очередь Queue. Дополнительно перегрузить следующие операции: / - добавить элемент; ++ - извлечь элемент; false - проверка, на содержание четных элементов в очереди; явный int() – количество положительных элементов в очереди</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Выделение первого числа, содержащегося в строке 2) Обнуление отрицательных элементов очереди
Вариант 23	<p>Класс - Строка. Дополнительно перегрузить следующие операции: < - сравнения строк по длине слов; + добавления числа к строке; - удаление последнего символа в строке; * – замена всех символов в строке на заданный.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Проверка наличия в строке служебных символов 2) Удаление знаков препинания из строки
Вариант 24	<p>Класс - Строка. Дополнительно перегрузить следующие операции: < - удаление всех символов равных заданному; + удаление нечетных символов; != сравнение длин строк; true - проверка на знаки препинания</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Проверка наличия в строке заданных символов 2) Удаление чисел их строки
Вариант 25	<p>Класс – Строка. Дополнительно перегрузить следующие операции: - – удалить элемент из строки из заданной позиции (типа string-item); > – проверка на вхождение подстроки; != – проверка строк на неравенство; + – добавить элемент в строку на заданную позицию (типа string+item).</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Подсчет количества слов в строке 2) Добавление в строке смайликов
Вариант 26	<p>Класс – Пароль. Дополнительно перегрузить следующие</p>

	<p>операции: - – замена последнего символа (типа password-item); > – сравнение длин паролей; != – проверка паролей на неравенство; ++ – сброс пароля на значение по умолчанию; true - проверка на стойкость.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> 1) Выделение среднего символа строки 2) Проверка допустимой длины пароля (6-12)
--	--

Вопросы

1. Перечислите свойства внутренних и вложенных классов?
2. Что такое статический класс и какие у него свойства?
3. Каково назначение перегрузки операторов?
4. Как используется ключевое слово operator?
5. Какие операции нельзя перегружать в C#?
6. Можно ли перегрузкой отменить очередность выполнения операции?
7. Истинно ли следующее утверждение: операция >= может быть перегружена.
8. Сколько аргументов требуется для определения перегруженной унарной операции?
9. Можно ли перегружать операцию []?
10. Можно ли перегружать операцию ->?
11. Приведите пример оператора приведения типа
12. Что такое метод расширения? Как и где его можно использовать?
13. Пусть дан фрагмент кода определения оператора преобразования типа. Определить форму преобразования.

```
public static implicit operator Point2D(Point3D a)
{ /* код */ }
```

14. Выберите верное утверждение. Метод расширения может:

- 1) получать доступ к public членам расширяемого класса
- 2) получать доступ к protected членам расширяемого класса
- 3) получать доступ к internal членам расширяемого класса
- 4) быть объявлен в любом классе
- 5) быть без параметров

15. Выберите из списка неверное правило перегрузки операторов для C#.

- 1) префиксные операции ++ и -- перегружаются парами
- 2) операции сравнения перегружаются парами: == и != ; < и >; <= и >=
- 3) перегруженные операции обязаны возвращать значения
- 4) должны объявляться как protected

5) true и false можно перегружать