

## № 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 MathOperation, содержащий 3 метода для работы с вашим классом (по варианту п.1): поиск максимального, минимального, подсчет количества элементов.
- 5) Добавьте к классу MathOperation методы расширения для типа string и вашего типа из задания №1. См. задание по вариантам.

<b>Вариант 1</b>	<p>Класс – <b>Одномерный массив</b>. Дополнительно перегрузить следующие операции: * – умножение массивов; true – истина если массив не содержит отрицательных элементов, int() – операция приведения – возвращает размер массива; == – проверка на равенство; &lt; – сравнение.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"><li>1) Проверка на содержание определённого символа в строке</li><li>2) Удаление отрицательных элементов</li></ol>
<b>Вариант 2</b>	<p>Класс – <b>Одномерный массив</b>. Дополнительно перегрузить следующие операции: - – разность со скалярным значением; &gt; – проверка на вхождение элемента; != – проверка на неравенство массивов ; + – объединение массивов</p> <p>Методы расширения:</p> <ol style="list-style-type: none"><li>1) Удаление гласных из строки</li><li>2) Удаление первых пяти элементов</li></ol>
<b>Вариант 3</b>	<p>Класс – множество <b>Set</b>. Дополнительно перегрузить следующие операции: + – 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"> <li>1) Добавление занятой после каждого слова</li> <li>2) Удаление повторяющихся из m множества</li> </ol>
<b>Вариант 4</b>	<p>Класс – множество <b>Set</b>. Дополнительно перегрузить следующие операции: - – удалить элемент из множества (типа set-item); * – пересечение множеств; &lt; – сравнение множеств; &gt; – проверка на подмножество; &amp; – придумайте использование.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> <li>1) Добавление точки в конце строки</li> <li>2) Удаление нулевых элементов из множества</li> </ol>
<b>Вариант 5</b>	<p>Класс – множество <b>Set</b>. Дополнительно перегрузить следующие операции: -&gt;&gt; – удалить элемент из множества (типа set-item); &gt; – проверка на подмножество; != – проверка множеств на неравенство; &lt;&lt; – добавить элемент в множество (типа set+item); % – пересечение множеств.</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> <li>1) Поиск самого короткого слова</li> <li>2) Упорядочивание множества</li> </ol>
<b>Вариант 6</b>	<p>Класс – однонаправленный список <b>List</b>. Дополнительно перегрузить следующие операции: &gt;&gt; – 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"> <li>1) Поиск самого длинного слова</li> <li>2) Удаление последнего элемента из списка</li> </ol>
<b>Вариант 7</b>	<p>Класс – множество <b>Set</b>. Дополнительно перегрузить следующие операции: ++ -добавление случайного элемента к множеству, + – объединение множеств; &lt;= – сравнение множеств; неявный int() – мощность множества; % - доступ к элементу в заданной позиции.</p>

	<p>Методы расширения:</p> <ol style="list-style-type: none"> <li>1) Шифрование строки</li> <li>2) Проверка на упорядоченность множества</li> </ol>
<b>Вариант 8</b>	<p>Класс – множество <b>Set</b>. Дополнительно перегрузить следующие операции: &gt; – проверка на принадлежность (типа операции in множественного типа)* – пересечение множеств; &lt; – проверка на подмножество; явный Date()– приведение типа</p> <p>Методы расширения:</p> <ol style="list-style-type: none"> <li>1) Выделение первого числа, содержащегося в строке</li> <li>2) Удаление положительных элементов из множества</li> </ol>
<b>Вариант 9</b>	<p>Класс – однонаправленный список <b>List</b>. 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"> <li>1) Выделение последнего числа, содержащегося в строке</li> <li>2) Удаление заданного элемента из списка</li> </ol>
<b>Вариант 10</b>	<p>Класс – список <b>List</b>. 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"> <li>1) Подсчет количества слов с заглавной буквы</li> <li>2) Проверка на повторяющиеся элементы в списке</li> </ol>
<b>Вариант 11</b>	<p>Класс – список <b>List</b>. 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"> <li>1) Подсчет количества слов.</li> <li>2) Проверка на нулевые элементы в списке</li> </ol>
<b>Вариант 12</b>	<p>Класс - однонаправленный список <b>List</b>. Дополнительно</p>

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

<b>Вариант 17</b>	<p>Класс - <b>Марица</b>. 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"> <li>1) Сумма целых чисел в строке</li> <li>2) Проверка на квадратность матрицы</li> </ol>
<b>Вариант 18</b>	<p>Класс - <b>Марица</b>. 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"> <li>1) Выделение первого числа, содержащегося в строке</li> <li>2) Обнуление отрицательных элементов матрицы</li> </ol>
<b>Вариант 19</b>	<p>Класс - <b>Марица</b>. Дополнительно перегрузить следующие операции: &lt; - 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"> <li>1) Разность двух первых целых чисел строки</li> <li>2) Сумма элементов матрицы</li> </ol>
<b>Вариант 20</b>	<p>Класс - стек <b>Stack</b>. 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"> <li>1) Подсчет количества вопросительных предложений в строке</li> <li>2) Проверка на ноль первого элемента</li> </ol>
<b>Вариант 21</b>	<p>Класс - стек <b>Stack</b>. Дополнительно перегрузить следующие операции: - - извлечение всех элементов равных заданному; ++ - дублирование верхнего элемента; &lt; копирование неповторяющихся элементов из второго стека.</p> <p>Методы расширения:</p>

	<ol style="list-style-type: none"> <li>1) Подсчет количества восклицательных предложений в строке</li> <li>2) Проверка стека на наличие отрицательных элементов</li> </ol>
<b>Вариант 22</b>	<p>Класс - очередь <b>Queue</b>. 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"> <li>1) Выделение первого числа, содержащегося в строке</li> <li>2) Обнуление отрицательных элементов очереди</li> </ol>
<b>Вариант 23</b>	<p>Класс - <b>Строка</b>. Дополнительно перегрузить следующие операции: &lt; - 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"> <li>1) Проверка наличия в строке служебных символов</li> <li>2) Удаление знаков препинания из строки</li> </ol>
<b>Вариант 24</b>	<p>Класс - <b>Строка</b>. Дополнительно перегрузить следующие операции: &lt; - 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"> <li>1) Проверка наличия в строке заданных символов</li> <li>2) Удаление чисел их строки</li> </ol>
<b>Вариант 25</b>	<p>Класс – <b>Строка</b>. Дополнительно перегрузить следующие операции: - – удалить элемент из строки из заданной позиции (типа string-item); &gt; – 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"> <li>1) Подсчет количества слов в строке</li> <li>2) Добавление в строке смайликов</li> </ol>
<b>Вариант 26</b>	<p>Класс – <b>Пароль</b>. Дополнительно перегрузить следующие</p>

	<p>операции: - – замена последнего символа (типа password-item); &gt; – 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"> <li>1) Выделение среднего символа строки</li> <li>2) Проверка допустимой длины пароля (6-12)</li> </ol>
--	--

## Вопросы

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

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

12. 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

13. 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