Контрольные вопросы:

- ☐ (5 б.) Перечислите все специальные функции-члены класса, включая перемещающие операции.
- ☐ (5 б.) Приведите примеры операторов, которые можно, нельзя и не рекомендуется перегружать.
- ☐ (5 б.) О каких преобразованиях следует помнить при проектировании операторов?
- ☐ (5 б.) Опишите классификацию выражений на основе перемещаемости и идентифицируемости.
- ☐ (5 б.) Зачем нужны rvalue-ссылки?
- ☐ (5 б.) Почему семантика перемещения лучше копирования?
- ☐ (5 б.) Что делает функия `std::move` и когда нет необходимости явно ее вызывать?
- ☐ (5 б.) Кем выполняется непосредственная работа по перемещению?
- ☐ (5 б.) Когда может потребоваться пользовательская реализация специальных функций-членов класса?
- ☐ (5 б.) Для чего нужны ключевые слова default и delete в объявлении специальных функций-членов класса?

Упражнения:

- ☐ (25 б.) Создайте класс, в котором будет 3 члена-данных: встроенный динамический массив чисел, переменная, хранящая длину данного массива, и контейнер `std::vector` чисел. Реализуйте для данного класса все специальные функции-члены. Для первых двух членов-данных все будет аналогично реализации класса String с семинара. Для вектора при перемещении необходимо будет использовать функцию `std::move` следующим образом: `v = std::move(other.v);` Напоминаю, функция `std::move` только маскирует объект, она выполняет приведение своего аргумента к типу rvalue-ссылки, позволяя тем самым разрешить перегрузку оператора присваивания в пользу перемещающей версии. У вектора, а также у всех остальных контейнеров из стандартной библиотеки, имеются перемещающие операции, которыми можно воспользоваться – для этого достаточно подготовить rvalue-ссылку. В конструкторах задействуйте по максимуму списки инициализации. В main напишите тестовый код, демонстрирующий использование специальных-функций членов.
- ☐ (50+ б.) Спроектируйте и реализуйте класс, представляющий дроби и средства для работы с ними. В качестве данных-членов должны выступать два числа – числитель и знаменатель. Для указания знака дроби можно использовать знаковый целочисленный тип для представления числителя. Реализуйте операторы ввода-вывода, сложения, вычитания, умножения, деления, сравнения. Реализуйте операторы вида op, например, +, как свободные функции и операторы вида op=, например, +=, как функции-члены класса. Также дополнительно реализуйте операторы префиксного и постфиксного инкремента и декремента как функции-члены класса, их описание Вы можете посмотреть в этой статье. Обязательно реализуйте функцию-член, выполняющую сокращение дроби и не забудьте вызывать ее после каждой операции. Специальные функции-члены в данном классе реализовывать самостоятельно не нужно, Вам будет достаточно версий, сгенерированных компилятором по умолчанию. Создайте необходимый набор конструкторов, сделайте оператор приведения дроби к типу double. Грамотно используйте explicit. Добавьте геттеры. Не забудьте про пространство имен и разбиение кода на файлы. Я буду оценивать качество реализации Вашего класса. В условии я перечислил много различных компонент. Не обязательно реализовывать все. Если реализуете все – можете дополнительно добавить что-то на свое усмотрение в пределах разумного. За данную задачу в зависимости от количества качественно реализованных компонент можно набрать более 50 баллов.