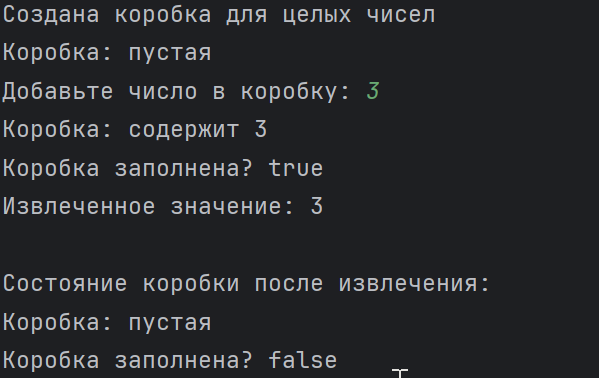
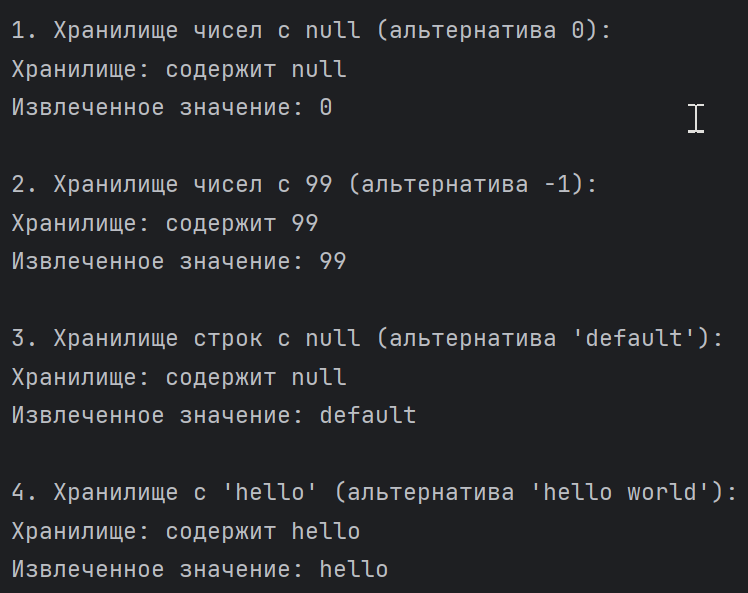
**Лабораторная работа №4**

**Задание 1.1. Коробка  
Текст задания:** Создайте сущность Коробка, которая обладает следующими характеристиками:  
• Может хранить один произвольный объект в один момент времени.  
• Объект можно получить и разместить на хранение в любой момент времени.  
• Если объект забирают из коробки – ссылку на этот объект необходимо обнулить.  
• Если объект кладут в коробку, но она не пуста – необходимо выкинуть исключение.  
• Имеет метод проверки на заполненность.  
• Методы класса должны работать с тем типом данных, который был указан во время создания объекта.  
Создайте Коробку, которая может хранить целочисленное значение, разместите туда число 3. Передайте Коробку в какой-либо метод, извлеките значение, и выведите его на экран.  
**Описание:  
Класс Box<T>** является обобщенным контейнером, предназначенным для хранения ровно одного объекта любого заданного типа T.  
**Приватные поля:**  
private T item - переменная для хранения объекта в коробке. Изначально имеет значение null.  
**Конструкторы:**  
public Box() - конструктор по умолчанию, который инициализирует коробку как пустую (item = null).  
**Публичные методы:  
public void put(T item)** - метод для помещения объекта в коробку. Сначала проверяет, не заполнена ли коробка (если this.item != null). Если коробка заполнена, выбрасывает исключение IllegalStateException с сообщением "Коробка уже заполнена". Если коробка пустая, присваивает переданный объект полю this.item.  
**public T take()** - метод для извлечения объекта из коробки. Сначала проверяет, не пуста ли коробка (если this.item == null). Если коробка пустая, выбрасывает исключение IllegalStateException с сообщением "Коробка пустая". Если в коробке есть объект, сохраняет его во временную переменную, обнуляет поле this.item и возвращает сохраненный объект.  
**public boolean isFull()** - метод проверки состояния коробки. Возвращает true, если в коробке есть объект, и false в противном случае.  
**public String toString()** - возвращает строковое представление коробки. Если коробка пуста, возвращает "Коробка: пустая", иначе возвращает "Коробка: содержит " + строковое представление хранимого объекта.  
Тестирование:

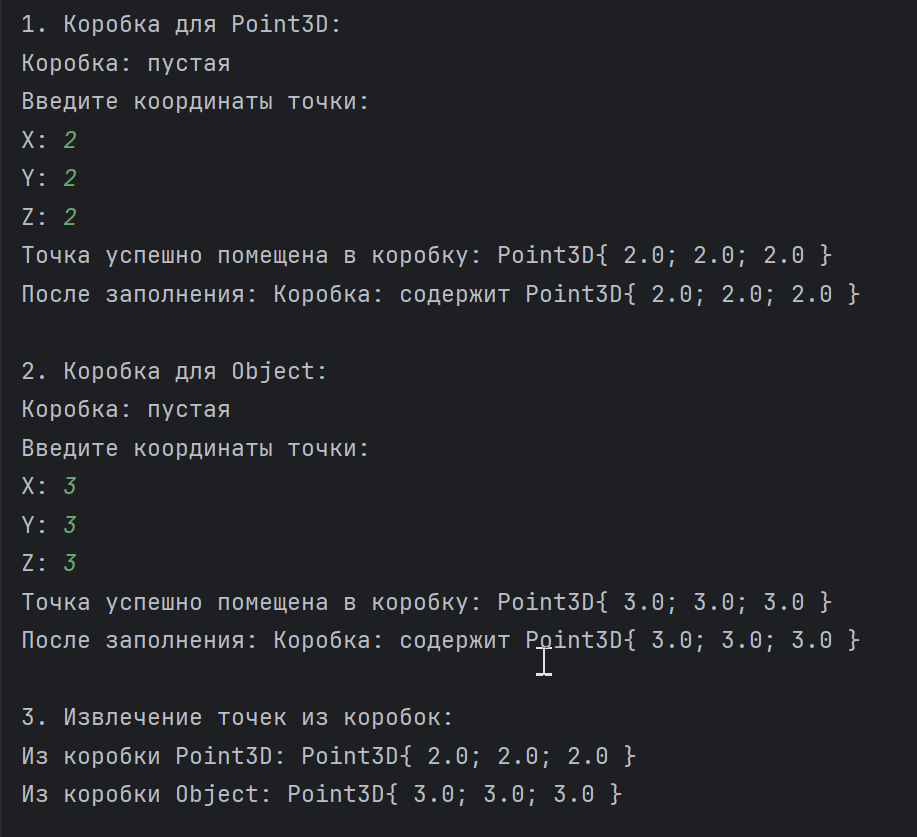


**Задание 1.2. Без null**  
**Текст задания:** Создайте сущность Хранилище, которая обладает следующими характеристиками:  
• Может хранить один произвольный объект в один момент времени.  
• Хранилище неизменяемо.  
• Объект кладется в Хранилище при его создании. В качестве объекта может быть сохранено также и значение null.  
• Хранилище может вернуть ссылку на Объект.  
• Если вместо объекта хранится null, необходимо вернуть какое-либо альтернативное значение.  
• Метод получения значения должен работать с тем типом данных, который был указан во время создания объекта. Выполните следующие задания:  
• Создайте Хранилище чисел, положите туда значение null. Передайте Хранилище в какойлибо метод, извлеките значение, и выведите его на экран. Альтернативой должно быть число 0.  
• Создайте Хранилище чисел, положите туда значение 99. Передайте Хранилище в какойлибо метод, извлеките значение, и выведите его на экран. Альтернативой должно быть число -1.  
• Создайте Хранилище строк, положите туда значение null. Передайте Хранилище в какойлибо метод, извлеките значение, и выведите его на экран. Альтернативой должна быть строка “default”. • Создайте Хранилище строк, положите туда значение “hello”. Передайте Хранилище в какой-либо метод, извлеките значение, и выведите его на экран. Альтернативой должна быть строка “hello world”  
**Описание:  
Класс Storage<T>** является обобщенной сущностью для хранения одного объекта с возможностью предоставления альтернативного значения.  
**Приватные поля:**  
private final T item - неизменяемое поле для хранения объекта.  
**Конструкторы:**  
public Storage(T item) - конструктор, который инициализирует поле item переданным значением.  
**Публичные методы:**  
public T get(T alternative) - метод для получения хранимого объекта. Проверяет значение поля item: если оно не равно null, возвращает item; если равно null, возвращает переданное альтернативное значение alternative.  
public String toString() - возвращает строковое представление хранилища. Если item равен null, возвращает "Хранилище: содержит null", иначе возвращает "Хранилище: содержит " + строковое представление хранимого объекта.  
**Тестирование:**



**Задание 2.3. Начало отсчета.**  
**Текст задания**: Создайте метод, принимающий Коробку из задачи 3.1.1, и кладет в неё трехмерную точку координат (из задачи 2.1.5) с произвольными значениями. Метод должен позволять передавать Коробку с более чем одним видом параметризации.  
**Описание:**  
Класс BoxUtils содержит метод для работы с коробками, способными хранить трехмерные точки.  
Публичные методы:  
public static void putPointInBox(Box<? super Point3D> box) - статический метод, который помещает объект Point3D в переданную коробку. Параметр метода использует нижнюю границу (? super Point3D), что позволяет передавать коробки, параметризованные типом Point3D или любым его суперклассом (например, Object).  
**Алгоритм работы метода:**

1. Запрашивает у пользователя ввод координат X, Y, Z с помощью InputValidation.inputDouble()
2. Создает новый объект Point3D с введенными координатами
3. Вызывает метод put() у переданной коробки, передавая созданную точку
4. Если коробка уже заполнена, выбрасывает исключение IllegalStateException.
5. При успешном помещении точки выводит сообщение о результате



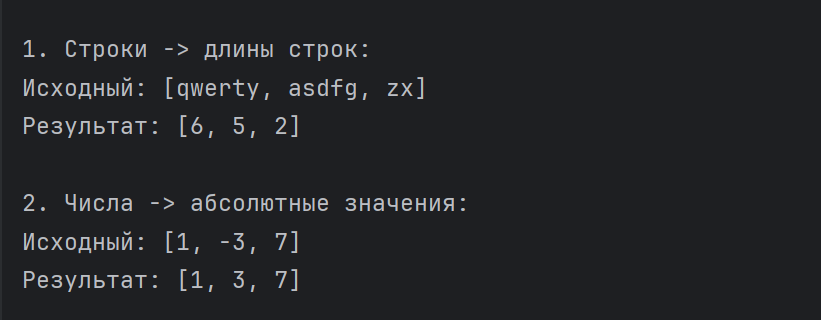
**Задание 3.1. Функция**  
**Текст задания:** Разработайте такой метод, который будет принимать список значений типа T, и объект имеющий единственный метод apply. Данный метод надо применить к каждому элементу списка, и вернуть новый список значений типа P, при этом типы T и P могут совпадать, а могут не совпадать. Используйте разработанный метод следующим образом:

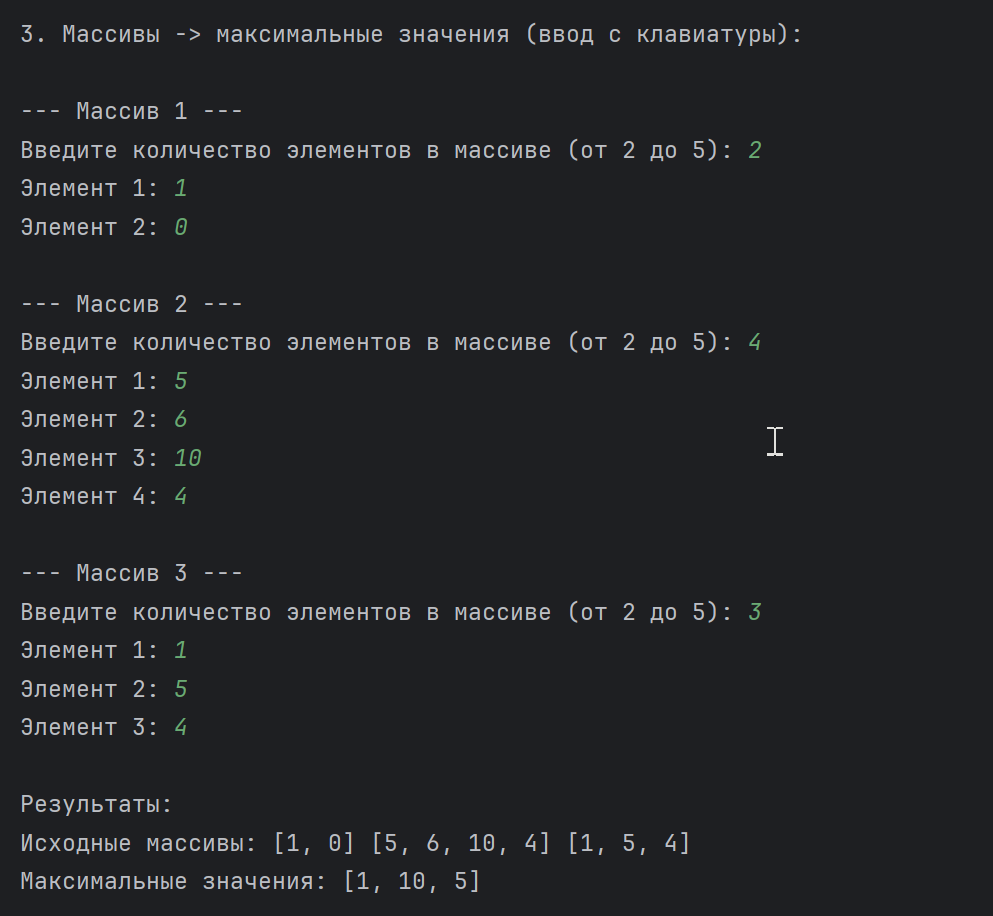
1. Передайте в метод список со значениями:“qwerty”, “asdfg”, “zx”, а получите список чисел, где каждое число соответствует длине каждой строки.
2. Передайте в метод список со значениями: 1,-3,7, а получите список в котором все отрицательные числа стали положительными, а положительные остались без изменений
3. Передайте в метод список, состоящий из массивов целых чисел, а получите список, в котором будут только максимальные значения каждого из исходных массивов  
   **Описание:  
   Класс ForTransformList**  
   Публичные методы:  
   transformList - метод для преобразования элементов списка с помощью функционального интерфейса. Принимает список и функциональный объект. Создает новый пустой список для результатов. Проходит по всем элементам исходного списка, применяет к каждому элементу переданную функцию объекта и добавляет результат в новый список. Возвращает заполненный список с результатами.  
   Логика работы в Main:  
   Создается три разных функциональных объекта для преобразования:

1.Для строк - вычисляет длину каждой строки

2.Для чисел - находит абсолютное значение каждого числа

3. Для массивов - находит максимальное значение в каждом массиве  
Для строк и чисел используются готовые списки. Для массивов организуется ввод с клавиатуры: запрашивается количество массивов, затем для каждого массива запрашивается количество элементов и значение каждого элемента. Все введенные массивы добавляются в список.  
Вызывается метод transformList для каждого случая, передавая соответствующий список и функциональный объект. Выводятся исходные данные и результаты преобразования.  
Тестирование:





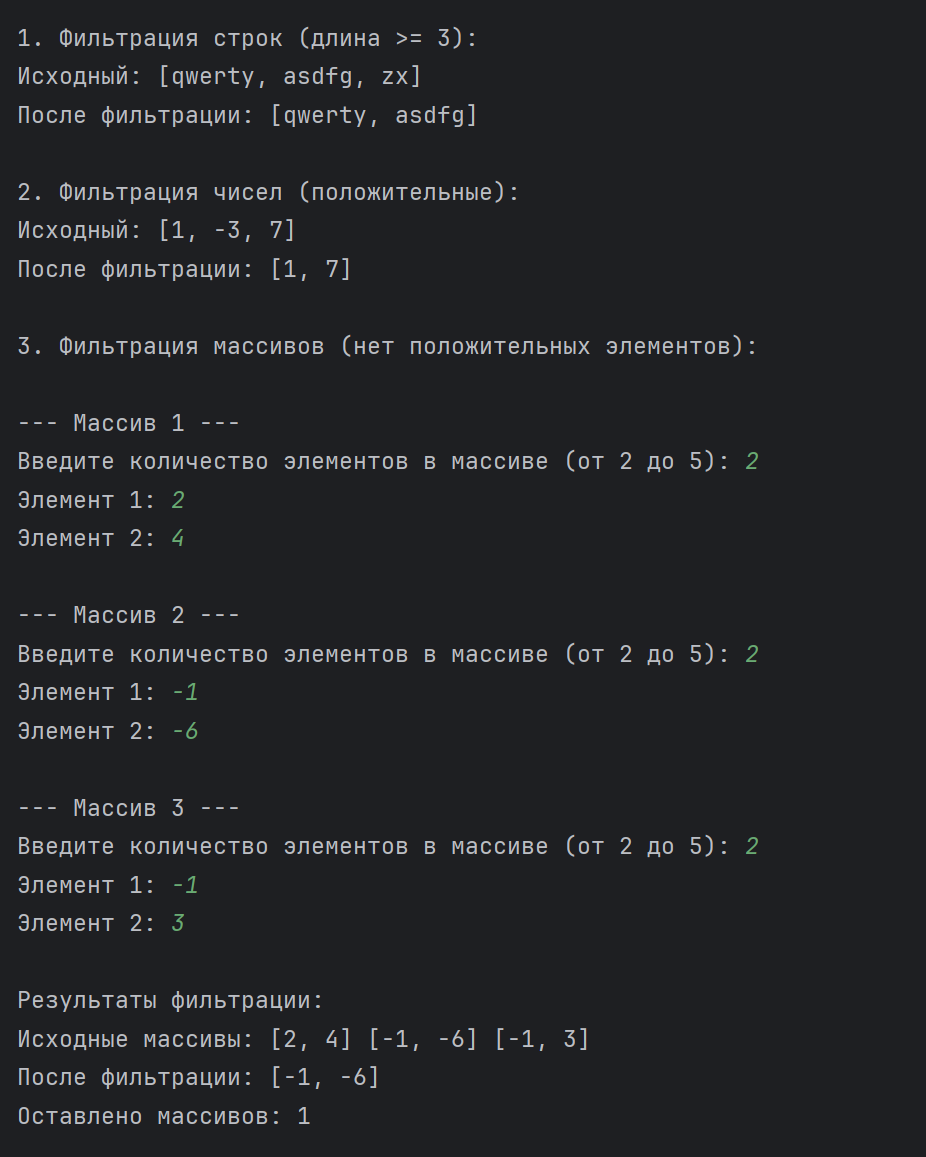
**Задание 3.2. Фильтр  
Текст задания**: Разработайте такой метод, который будет принимать список значений типа T и объект имеющий единственный метод test (принимает T и возвращает boolean). Верните новый список типа T, из которого удалены все значения не прошедшие проверку условием. Используйте разработанный метод следующим образом:

1. Передайте в метод список со значениями: “qwerty”, “asdfg”, “zx”, и отфильтруйте все строки имеющие менее трех символов
2. Передайте в метод список со значениями: 1,-3,7, и отфильтруйте все положительные элементы
3. Передайте в метод список состоящий из массивов целых чисел, а получите список в котором будут только те массивы, в которых нет ни одного положительного элемента  
   **Описание:  
   Класс Filter**Публичные методы:  
   • filterList - метод для фильтрации элементов списка по условию. Принимает список и предикат Predicate<T>. Создает новый пустой список для результатов. Проходит по всем элементам исходного списка, проверяет каждый элемент с помощью переданного предиката. Если элемент удовлетворяет условию, добавляет его в новый список. Возвращает список с отфильтрованными элементами.  
     
   Логика работы в Main:  
   Создается три разных предиката для фильтрации:

1.Предикат для строк - проверяет что длина строки не менее 3 символов

2.Предикат для чисел - проверяет что число положительное

3.Предикат для массивов - проверяет что в массиве нет положительных элементов.  
Для строк и чисел используются готовые списки. Для массивов организуется ввод с клавиатуры: запрашивается количество массивов, затем для каждого массива запрашивается количество элементов и значение каждого элемента. Все введенные массивы добавляются в список.  
Вызывается метод filterList для каждого случая, передавая соответствующий список и предикат. Выводятся исходные данные и результаты фильтрации, а также количество оставшихся элементов.  
Тестирование:



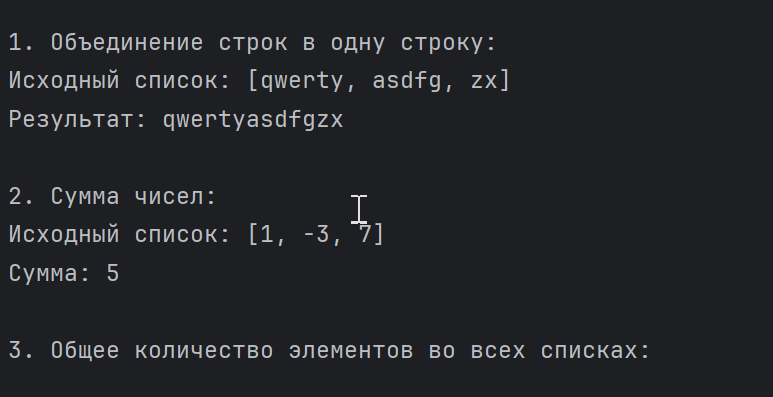
**Задание 3.3. Сокращение**  
**Текст задания:** Разработайте такой метод, который будет принимать список значений типа T и способ с помощью которого список значений можно свести к одному значению типа T, которое и возвращается из метода. Используйте разработанный метод следующим образом:

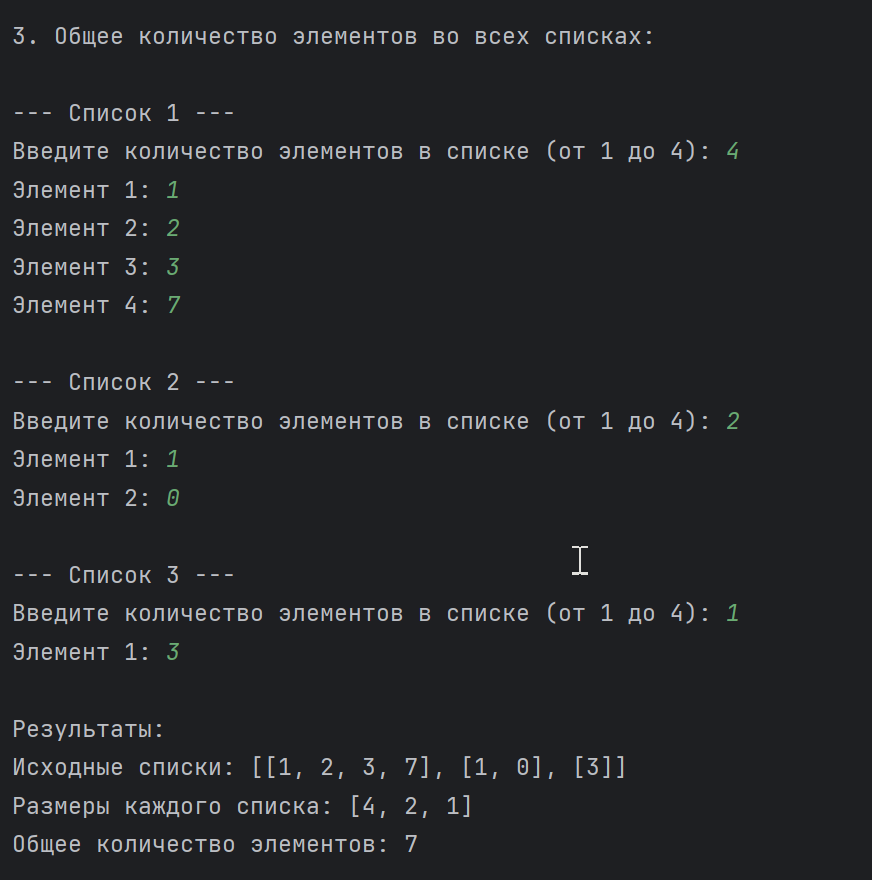
1. Передайте в метод список со значениями: “qwerty”, “asdfg”, “zx”, и сформируйте одну большую строку, которая состоит из всех строк исходного списка.
2. Передайте в метод список со значениями: 1,-3,7, и верните сумму всех значений исходного списка.
3. Имеется список, состоящий из списков целых чисел, получите общеe количество элементов во всех списках. Подсказка: решить задачу можно в одно действие или последовательно использовать методы из 3.3.1 и 3.3.3.  
   Далее необходимо изменить разработанный метод таким образом, чтобы данный метод гарантированно не возвращал null и не выбрасывал ошибок в том случае, если исходный список пуст.  
   **Описание:  
   Класс Reduce**Публичные методы:  
   • reduceList - метод для сокращения списка до одного значения. Принимает список, бинарный оператор BinaryOperator<T> и дефолтное значение. Если список пуст, возвращает дефолтное значение. Берет первый элемент списка как начальный результат. Затем проходит по оставшимся элементам списка, применяя операцию объединения к текущему результату и следующему элементу. Возвращает окончательный результат после обработки всех элементов.  
     
   Логика работы в Main:  
   Создается три разных бинарных оператора для сокращения:

1.Бинарный оператор для строк - объединяет строки в одну

2.Бинарный оператор для чисел - складывает числа

4.Бинарный оператор для подсчета элементов - суммирует размеры списков.  
Для строк и чисел используются готовые списки. Для работы со списками организуется ввод с клавиатуры: запрашивается количество списков, затем для каждого списка запрашивается количество элементов и значение каждого элемента. Все введенные списки добавляются в общий список.  
Сначала создается список размеров каждого внутреннего списка. Затем вызывается метод reduceList для списка размеров, передавая операцию сложения. Выводятся исходные данные, размеры каждого списка и общее количество элементов.  
Тестирование:





**Задание 3.4. Коллекции**  
**Текст задания:** Разработайте такой метод, который будет возвращать коллекцию типа P со значениями типа T. Данный метод будет принимать:

1. Список исходных значений
2. Способ создания результирующей коллекции
3. Способ передачи значений исходного списка в результирующую коллекцию.  
   Используйте разработанный метод следующим образом:
4. Передайте в метод список со значениями: 1,-3,7, и верните их разбитыми на два подсписка, в одном из которых будут только положительные числа, а в другом только отрицательные.
5. Передайте в метод список со значениями: “qwerty”, “asdfg”, “zx”, “qw” и верните их разбитыми на подсписки таким образом, чтобы в любом подсписке были строки только одинаковой длины
6. Передайте в метод список со значениями: “qwerty”, “asdfg”, “qwerty”, “qw” и верните набор такого вида, который не может содержать одинаковые объекты  
   **Описание:  
   Класс ForCollections**  
   Публичные методы:  
   • collectToList - метод для преобразования списка в указанный тип коллекции с фильтрацией. Принимает исходный список, поставщик Supplier<Collection<T>> для создания коллекции и предикат Predicate<T> для включения элементов. Создает новую коллекцию с помощью поставщика. Проходит по всем элементам исходного списка, проверяет каждый элемент с помощью предиката. Если элемент удовлетворяет условию, добавляет его в коллекцию. Возвращает заполненную коллекцию.  
   **Тестирование:**

