Zadanie 1.

Napisz klasę szablonową CustomList implementującą strukturę listy jednokierunkowej ze wskaźnikami na początek i koniec. Zaprogramuj w niej metody:

- void addLast(T value) - dodającą wartość na koniec listy,
- T getLast() - zwracającą wartość z końca listy,
- void addFirst(T value) - dodającą wartość na początek listy,
- T getFirst() - zwracającą wartość z początku listy,
- T removeFirst() - zwracającą i usuwającą element z początku listy,
- T removeLast() - zwracającą i usuwającą element z końca listy.

Zadanie 2.

Niech klasa CustomList dziedziczy po klasie AbstractList. Wygeneruj potrzebne metody. Nadpisz i zaprogramuj metody:

- boolean add(T t) - działającą tak samo jak addLast i zwracającą prawdę,
- int size() - zwracającą rozmiar listy,
- T get(int index) - zwracającą wartość w węźle o podanym indeksie.

Zadanie 3.

Nadpisz i zaprogramuj metody:

- Iterator<T> iterator() - zwracającą iterator do listy. Zdefiniuj w niej iterator,

- Stream<T> stream() - zwracającą strumień z zawartością listy.

Zadanie 4.

Napisz statyczną metodę szablonową, która przyjmie jako parametry Listę obiektów typu szablonowego T oraz obiekt Class. Metoda powinna zwrócić listę obiektów, które należą do wskazanej klasy.

Następnie zmodyfikuj metodę tak, aby filtrowała obiekty, które dziedziczą (bezpośrednio lub pośrednio) po wskazanej klasie.

Zadanie 5.

Napisz predykat, który porówna, czy testowana zmienna znajduje się w otwartym przedziale, zdefiniowanym jego granicami.

Korzystając z niego napisz metodę statyczną, która dla listy oraz granic zakresu danych typem szablonowym zwróci liczbę elementów w tej liście, spełniających warunek predykatu.

Zadanie 6.

Napisz komparator, który porówna dwie kolekcje pod względem liczby ich elementów. Następnie zmodyfikuj go tak, aby przyjmował wyłącznie kolekcje liczb i porównywał je pod względem ich sumy.