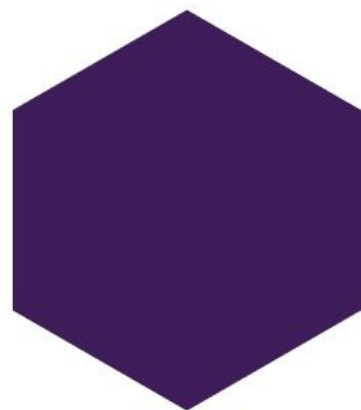
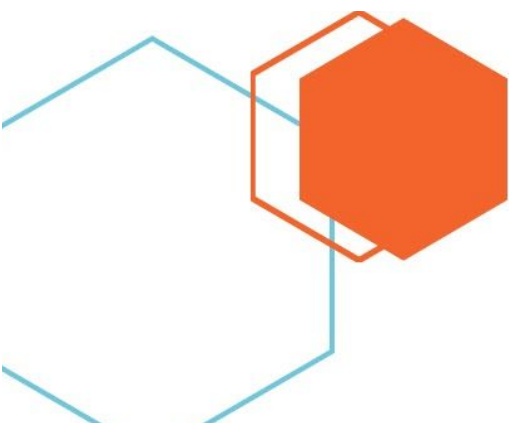




Аннотация: этот документ является практическим руководством к Week 2 Day 3 Coding Bootcamp.

Академия Ковалевского





Задания: List

3

Класс List<T>

3





Задания: List

Класс List<T>

Написать свой класс List, в котором сделать все следующие методы. Объект класса можно создавать только при помощи **new**.

```
package com.kovalevskiyi.academy.codingbootcamp.week2.day3;

import java.util.function.Function;

public class List<T> {

    private List(List<T> prev, List<T> next, T value) {

        // TODO

    }

    public static <T> List<T> createOne(T value) {

        // TODO

    }

    public static <T> List<T> addToEnd(List<T> list, T newValue) {

        // TODO

    }

    public static <T> List<T> addToStart(List<T> list, T newValue) {

        // TODO

    }

    public static <T> boolean contains(List<T> list, T value) {

        // TODO

    }

    public static <T, R> List<R> map(List<T> list, Function<T, R> mapFunction) {

        // TODO

    }

}
```



```
public int length() {  
    // TODO  
}  
  
public List<T> insertAfter(T newValue) {  
    // TODO  
}  
  
public void insertAfter(T[] newValues) {  
    // TODO  
}  
  
public void delete() {  
    // TODO  
}  
  
public void connect(List<T> to) {  
    // TODO  
}  
  
public List<T> getNext() {  
    // TODO  
}  
  
public List<T> getPrev() {  
    // TODO  
}  
  
public void setPrev(List<T> prev) {  
    // TODO  
}  
  
public void setNext(List<T> next) {  
    // TODO  
}
```





```
public T getValue() {  
    // TODO  
}  
}
```

List(List<T> prev, List<T> next, T value)

```
private List(List<T> prev, List<T> next, T value) {  
    // TODO  
}
```

Этот конструктор на вход принимает предыдущий и следующий элементы листа, и текущее значение.

createOne(T value)

```
public static <T> List<T> createOne(T value) {  
    // TODO  
}
```

Написать метод **createOne()**, который создает новый элемент с значением value.

addToEnd(List<T> list, T newValue)

```
public static <T> List<T> addToEnd(List<T> list, T newValue) {  
    // TODO  
}
```

Написать метод **addToEnd()**, который добавляет новое значение в конец листа.



addToStart(List<T> list, T newValue)

```
public static <T> List<T> addToStart(List<T> list, T newValue) {  
    // TODO  
}
```

Написать метод **addToStart()**, который добавляет новое значение в начало листа.

contains(List<T> list, T value)

```
public static <T> boolean contains(List<T> list, T value) {  
    // TODO  
}
```

Написать метод **contains()**, которой возвращает true, если заданное значение присутствует в листе и false, если такого значения нет.

map(List<T> list, Function<T, R> mapFunction)

```
public static <T, R> List<R> map(List<T> list, Function<T, R> mapFunction) {  
    // TODO  
}
```

Написать метод **map()**, который создает новый лист равный по размеру текущему листу, в котором каждый элемент получен путем применения mapFunction к элементу текущего листа.

length()

```
public int length() {  
    // TODO  
}
```

Написать метод **length()**, который возвращает количество элементов в текущем листе.





insertAfter(T newValue)

```
public List<T> insertAfter(T newValue) {  
    // TODO  
}
```

Написать метод **insertAfter()**, который добавляет новое значение после текущего значения листа.

insertAfter(T[] newValues)

```
public void insertAfter(T[] newValues) {  
    // TODO  
}
```

Написать метод **insertAfter()**, который добавляет новые значения после текущего значения листа.

delete()

```
public void delete() {  
    // TODO  
}
```

Написать метод **delete()**, который удаляет текущий элемент листа.

connect(List<T> to)

```
public void connect(List<T> to) {  
    // TODO  
}
```

Написать метод **connect()**, который связывает текущий элемент листа с входящим **to**.





getNext()

```
public List<T> getNext() {  
    // TODO  
}
```

Написать метод **getNext()**, который возвращает следующий элемент листа.

getPrev()

```
public List<T> getPrev() {  
    // TODO  
}
```

Написать метод **getPrev()**, который возвращает предыдущий элемент листа.

setPrev(List<T> prev)

```
public void setPrev(List<T> prev) {  
    // TODO  
}
```

Написать метод **setPrev()**, который устанавливает предыдущий элемент текущему элементу листа.

setNext(List<T> next)

```
public void setNext(List<T> next) {  
    // TODO  
}
```

Написать метод **setNext()**, который устанавливает следующий элемент текущему элементу листа.





getValue()

```
public T getValue() {  
    // TODO  
}
```

Написать метод **getValue()**, который возвращает значение текущего элемента листа.

