תרגיל Generics – Linked list

בתרגיל זה נשדרג את המחלקה של הרשימה המקושרת המופיעה במצגת (ראה 2 מחלקות בדף הבא). להזכירך רשימה מקושרת הינה רשימה משורשרת של איברים (מסוג ListItem). כל איבר כזה מחזיר אובייקט מידע מסוג T ומצביע לאיבר הבא ברשימה. האיבר האחרון ברשימה מצביע ל null. ראה סרטוט מטה המדגים שרשור של שלושה איברים.

1. עליך להשתמש בקוד הקיים ולהוסיף את השיטות הבאות למחלקה List:
2. void clear()

הופך את הרשימה לרשימה ריקה.

1. int size()

מחזיר את מספר אברי הרשימה.

1. boolean contains(T t)

מחזיר אמת אם t נמצא ברשימה, אחרת שקר.

1. void addLast(T t)

מוסיף את t לסוף הרשימה.

1. Object[] toArray()

מחזיר את האובייקטים ברשימה במערך מסוג Object לפי הסדר שלהם ברשימה.

1. T get (int index)

מחזיר את t במקום ה index ברשימה. אם יש חריגה מגבולות הרשימה

(index < 0 || index > size() (השיטה תזרוק את החריגה IndexOutOfBoundsException.

1. T getLast()

מחזיר את t האחרון ברשימה. אם הרשימה ריקה ייזרק NoSuchElementException.

1. int indexOf(T t)

מחזיר את האינדקס של המופע הראשון של t ברשימה או 1- אם t לא נמצא ברשימה.

1. boolean remove(T t)

מוריד את המופע הראשון של t ברשימה. מחזיר אמת אם t נמצא ברשימה ושקר אחרת.

1. עליך להדגים את השיטות שמימשת בשיטה main במחלקה Program.

head

null

T

T

T

**public** **class** ListItem<T> {

**private** ListItem<T> next;

**private** T info;

**public** ListItem(T info, ListItem next) {

**this**.info = info;

**this**.next = next;

}

**public** T getElm() {

**return** info;

}

}

**public** **class** List<T> {

**private** ListItem<T> head;

**public** List() {

head = **null**;

}

**public** **void** addItem(T info) {

head = **new** ListItem<T>(info, head);

}

**public** T firstElm() {

**return** head.getElm();

}

}