

תיקונים: 18.7.24

מבנה נתונים, תשפ"ד

מיני פרויקט תכנותי מסכם תור קדימויות מבוסס ערימה בינרית משקל 25% מהציון

יש להגיש את התרגיל בזוגות או בשלושות.

מטרות התרגיל

- תרגול מימוש מבנה נתונים דינמי.
- תרגול כתיבת קוד תקין, מתועד היטב לפי הסטנדרטים והקונבנציות המקובלים בתעשייה.
- תרגול פתרון בעיות ולמידה עצמית מעבר למה שנלמד בכיתה.
- תרגול מושגים בתכנות מונחה עצמים: ממשק (interface).

תיאור התרגיל

בתרגיל זה עליכם לממש מחלקת תור קדימויות באמצעות ערימה בינרית.

מחלקות/ממשקים נדרשים

public class MyPriorityQueue<T extends Comparable<T>> implements Queue

תיאור: מחלקה המייצגת תור קדימויות, הממומש באמצעות ערימה בינרית.

אלה המתודות הנדרשות:

הרשאה	ערך מוחזר	
public	MyPriorityQueue<T>	MyPriorityQueue() קונסטרוקטור. בונה תור ריק. גודל המערך ברירת מחדל למערך: 10. הערה: לצורך הגדרת ברירת מחדל יש להשתמש בקבוע ולא לקודד את הערך 10 לתוך המתודה.
public	MyPriorityQueue<T>	MyPriorityQueue(int size) קונסטרוקטור. בונה תור ריק עם מערך בגודל size.
public	String	toString() מחזירה ייצוג של התור (האיברים לפי סדר השליפה) מבלי לשנות את תוכן התור. רמז: להעתיק את המערך למערך עזר, למיין את מערך העזר, ולהחזיר את ה-toString של המערך.
public	static void	sort(T[] arr) ממיינת את מערך הקלט במיין ערימה. רמז: ליצור אובייקט חדש של תור קדימויות. להחליף לו את המערך במערך הקלט arr. לשנות לו את הגודל לגודל של arr. להפעיל עליו את אלגוריתם המיין (הוצאת איבר, הצבת האיבר שהוצאנו בסוף המערך, עד שהערימה ריקה). תיקון: המתודה לא static.

בנוסף, על המחלקה לממש את הממשק Queue של Java, כלומר לממש את המתודות הבאות:

	Throws exception	Returns special value
Insert	add(e)	offer(e)
Remove	remove()	poll()
Examine	element()	peek()

בנוסף, יש לממש את המתודות הבאות שהממשק Queue יורש מ-Collection:

- size()
- isEmpty()
- clear()
- contains()

אין צורך לממש מתודות נוספות של Queue.

אם מנסים לקרוא מתור ריק, המתודות remove() ו-element() זורקות חריגה מסוג NoSuchElementException.

שימו לב: המחלקה NoSuchElementException כבר קיימת. EmptyQueueException – שימו לב שהמחלקה

EmptyQueueException לא קיימת ועליכם לממש אותה כיורשת של המחלקה Exception.

לעומת זאת, אם מנסים לקרוא מתור ריק, המתודות poll() ו-peek() מחזירות null.

לא ניתן לאחסן ערכי null בתור. אם מנסים להכניס null לתור, המתודה add() זורקת NullPointerException (שימו לב: המחלקה הזאת כבר קיימת).

אם מנסים להכניס null לתור, המתודה offer() מחזירה false. אחרת – היא מחזירה true.

שימו לב: המחלקה MyPriorityQueue מניחה שהמחלקה T מממשת את הממשק Comparable של ג'אווה, כלומר: שאובייקט מסוג T מממש את המתודה compareTo(T y) שמחזירה את תוצאות ההשוואה בין שני אובייקטים מסוג T. המתודה compareTo מחזירה ערך קטן מאפס אם האובייקט הנוכחי קטן מהאובייקט השני, ערך גדול מאפס אם האובייקט הנוכחי גדול מהאובייקט השני, ואפס אם הם שווים. למשל, נניח שיש לנו שני אובייקטים x ו-y שמממשים את הממשק Comparable, אז במקום לכתוב (כמו שמשווים משתנים פרימיטיביים):

if (x < y) ❌

צריך לכתוב (שימו לב להמרה):

if (((Comparable) x).compareTo((Comparable) y) < 0) ✅

רמזים

- יש לשמור מערך של T כתכונה private. יש לשמור כתכונה private גם את גודל הערימה size.
- יש לממש את heapify(n) כמתודה private ולהשתמש בה במימוש שאר המתודות.
- גודל התור אינסופי. כדאי לכתוב מתודה פרטית שכאשר המערך מתמלא, היא יוצרת מערך חדש בגודל כפול מהקודם ומעתיקה אליו את האיברים של המערך הקודם.

דרישות – יורדו הרבה נקודות למי שלא עומד בדרישות גם אם הקוד עובד

- יש לכתוב את שמות המגישים בהערה בראש הקוד.
- חשוב לממש את הממשק בדיוק כמו שהוא כתוב. אין לשנות את שמות המתודות, ויש להקפיד במיוחד על אותיות גדולות/קטנות. שינוי שם המתודה אפילו באות אחת גורם לכך שקוד שמשמש במתודה, למשל הקוד שבודק את התרגיל, לא יעבוד.
- הקבצים חייבים להתקמפל.
- על הקוד לעבוד באופן תקין ללא שגיאות. בפרט, יש לטפל נכון בערכי null.
- יש להשתמש בשמות משתנים משמעותיים באנגלית.
- יש להקפיד על קונבנציות שימוש באותיות גדולות/קטנות לפי המקובל ב-Java (ראו כאן).
- יש לתעד את הקוד תיעוד מלא לפי הקונבנציות של Javadoc (ראו עמודים 68-70 בספר של ברנדס).
- כל זוג כותב בעצמו את הקוד שלו.
- אסור לשתף קבצים או קטעי קוד בין סטודנטים שאינם באותה קבוצה (זאת עבירת משמעת).
- אסור להעתיק קטעי קוד שלא אתם כתבתם, למשל קטעי קוד שמצאתם באינטרנט (זאת עבירת משמעת).
- מותר להשתמש בכלי AI כגון ChatGPT אבל יש לסמן בהערה את הקוד שהם יצרו.

הגשה

עליכם להגיש במודל את הקבצים הבאים :

MyPriorityQueue.java 📎

EmptyQueueException.java 📎

יש להגיש קבצים **בדיוק בשם הזה**, המכילים את הקוד. **אין** להגיש קובץ בשם אחר. אפשר בתוך zip אבל לארוז בו רק את הקבצים של התרגיל.