



## הנדסת תוכנה – תרגיל בית 4

### דגשים להגשת המטלה

1. **תאריך הגשה: יום חמישי 26.06.2025, בשעה 23:59.**
2. **הגשה בזוגות בלבד, אלא אם ניתן אישור!**
3. הקוד חייב להיכתב בהתאם למוסכמות כתיבת הקוד בקורס כולל תיעוד כנדרש. קוד שלא עומד בדרישות יגרור הורדת ניקוד. ניתן למצוא את קובץ מוסכמות הקידוד באתר הקורס תחת הלשונית "קבצי עזר".
4. ההגשה מתבצעת ב-Moodle באזור המיועד על ידי אחד מהשותפים, לאחר יצירת קבוצה.
5. ניתן להגיש את התרגיל לכל היותר עד 48 שעות לאחר מועד ההגשה ללא הורדת ניקוד. לאחר 48 שעות תיבת ההגשה תיסגר ולא יהיה ניתן להגיש את התרגיל כלל.
6. שאלות והבהרות ייכתבו בפורום ייעודי שייפתח לתרגיל הבית ב-Moodle.
7. עליכם לוודא את נכונות ההגשה, כולל הורדה של קובץ ה- zip מהמודל, פתיחתו והרצת הקבצים בסביבה נקייה. לא תתקבלנה טענות בנוגע להגשת קבצים לא נכונים. האחריות עליכם לוודא שההגשה אכן בוצעה ושהקבצים עלו למערכת המודל. לא תתקבלנה טענות בעניין זה.

### מטרת התרגיל

עבודה עם מבני נתונים, חריגות, העתקת עצמים ואיטרטורים.

### הכנות טרם תחילת התרגיל

1. פתיחת פרויקט Java חדש. הנחיות לפתיחת פרויקט והכנת סביבת העבודה ניתן למצוא [באתר](#) של JetBrains.
2. הורדת קבצי התרגיל.

### הוראות כלליות

1. מומלץ להריץ את התוכנית עם מספר קלטים שונים ולחשוב על מקרי קצה אפשריים.
2. מומלץ לחזור על התרגולים וההרצאות וכן להיעזר באינטרנט.
3. מומלץ להשתמש ב-Git במהלך כתיבת התרגיל.
4. שימו לב, העתיקו סימני גרש מקבצי הפלטים שפורסמו לכם בעבר, לא ממסמך זה.

### הוראות הגשה

1. הגשה אלקטרונית בלבד דרך אתר הקורס ב-moodle. ההגשה תכלול את קובץ ה- zip בלבד בפורמט HW4\_id1\_id2.zip כאשר id1, id2 הם תעודות הזהות של המגישים. אם בהגשה יש מספר שונה של סטודנטים תהיה הוספה או הורדה של מספר תעודות זהות בהתאם. על קובץ ה- zip להכיל תיקייה אחת בלבד בשם src, אשר תכיל את כלל קבצי הקוד אשר כתבתם (שימו לב, java. הוא סוג הקובץ), ללא הקובץ Main.
2. אין להגיש את הקבצים כחלק מ-package.
3. הגשה אלקטרונית בלבד דרך אתר הקורס ב-moodle. ההגשה תכלול את קובץ ה- zip בלבד.



הטכניון – מכון טכנולוגי לישראל  
הפקולטה למדעי הנדסה והחלשות  
הנדסת תוכנה אביב תשפ"ה



4. ההגשה מתבצעת על ידי אחד מבני הזוג לאחר שיצר קבוצה ובן הזוג השני הצטרף אליה.
5. תרגיל בית שלא יוגש על פי הוראות ההגשה – לא ייבדק ויגרור **ציון 0**.
6. יש להקפיד על יושרת הכנת התרגיל וההגשה.
7. יש לוודא כי הקוד מתקמפל – קוד אשר לא יעבור הידור יקבל ציון 0.
8. אין צורך להגיש את קבצי הקלט והפלט אשר ניתנים כחלק מתרגיל זה.



### מערכת כניסה לתיבת נוח:

נרצה לבנות מערכת אשר מנהלת את התור של החיות הנכנסות לתיבה של נוח. לתיבה יכולים להיכנס חמישה סוגים של חיות: אריה, נמר, קוף, נחש וזברה.

לכל חיה יש מספר שלם שמראה על הדומיננטיות שלה,

חיות עומדות בתור לכניסה לתיבה, כאשר מגיעה חיה חדשה, היא נכנס לתור לפי סדר הדומיננטיות (חיה עם דומיננטיות 3 נכנסת לפני חיה עם דומיננטיות 2 ו-1).

לצורך כך ניצור מחלקת מבנה נתונים גנרית חדשה.

### מחלקת SpeciesQueue

תור זנים הינו סוג של תור אשר נותן עדיפות לאלמנטים שונים. כאשר מוסיפים איבר חדש, מכניסים אותו לפי סדר העדיפות שלו. במידה ויש כמה סוגים של איברים עם אותו סדר עדיפות, נדאג שאיברים מאותה המחלקה יהיו ביחד, כאשר החיה החדשה שנכנסה תהיה ראשונה בקבוצה שלה (האינדקס הכי נמוך).

תור זנים מאותחל לגודל 10, ובכל פעם שנגמר המקום, נגדיל את התור פי 2.

עליכם לממש את המחלקה כמחלקה גנרית ובעזרת מערך בלבד. אין להשתמש במבני נתונים אשר מוגדרים ב-Java.

בנוסף, נרצה ליצור מחלקת חריגות SpeciesQueueException אשר תייצג חריגות בלתי מסומנות של המחלקה (זו לא מחלקה אבסטרקטית). עליכם ליצור שתי מחלקות חריגות אשר יירשו ממחלקה זו, אחת בשם InvalidInputException ו- EmptyQueueException. יש לכתוב את כלל הבנאים אשר הזכרנו.

תכונות המחלקה:

- מערך אשר יכיל את איברי התור.
- מספר האיברים בתור.

פעולות המחלקה:

- פעולת הוספה בשם add. הפעולה תקבל איבר חדש להכנסה ותכניס את האיבר לפי התיאור למעלה. אם האיבר שיש להוסיף הינו null יש לזרוק את החריגה InvalidInputException.
- פעולת הסרה בשם remove אשר תחזיר את ראש התור ותסיר אותו. במקרה בו התור ריק, יש לזרוק את החריגה EmptyQueueException.



- פעולה בשם peek אשר תחזיר את ראש התור ללא הסרתו. במקרה בו התור ריק יש לזרוק את החריגה `EmptyQueueException`.
- פעולה בשם size אשר תחזיר את מספר האיברים בתור.
- פעולת clone אשר תבצע העתקה עמוקה של התור. עליכם לדאוג לכך שיהיה ניתן להכניס איברים לתור רק אם הם ממשים את `Cloneable` (רמז: חשבו מה זה אומר על הטיפוס הגנרי). עליכם להשתמש ב `covariant return type` וכן בבלוק `try-catch`. במקרה של תפיסת חריגה יש להחזיר `null`. בכל מקרה, אין לזרוק חריגות מפעולה זו. תוכלו להיעזר בפעולות `getMethod` ו-`invoke` לצורך העתקת איברי תור. בנוסף, על המחלקה לממש את `Iterable`, כך שהאיטרטור שלה יחזיר איברים מהטיפוס הגנרי. האיטרטור יעבור על איברי התור לפי הסדר (מאינדקס 0 ועד לסוף)
- בנוסף, על איברי המחלקה לממש את `Comparable`, כך שבדיקה של איזה איבר הוא בעל קדימות בתור תתבצע על ידי השיטה `compareTo`.

### מחלקות החיות:

ישנם 5 זנים של חיות, בעלי רמות הדומיננטיות הבאות: Lion 4, Tiger, 4, Monkey 3, Snake 2, Zebra 1  
חשבו איך נכון לממש מחלקות אלו על מנת להשלים את הדרוש.

### מחלקת Ark

מחלקה זו תייצג את התיבה.

תכונות המחלקה:

- תור של חיות

פעולות המחלקה:

- בנאי ריק אשר מאתחל את התור.
- פעולה בשם add אשר מקבל חיה ומוסיפה אותה לתור.
- פעולה בשם `enterToArk` אשר תכניס לתיבה את החיה הראשונה בתור ותדפיס `"A <species> entered the ark"` כאשר `<species>` יוחלף בשם החיה. (שימו לב כי אין באמת צורך לשמור את החיות בתוך התיבה, הכנסה לתיבה משמעותה הוצאה מהתור).
- פעולה בשם `enterAllToArk` אשר תכניס את כל החיות לתיבה אחת אחת.



- פעולה בשם `showQueue` אשר תציג את התור באופן הבא:  
"`<species 1>, <species 2>, ..., <species n>`" כאשר אלו שמות החיות (שמות המחלקות, מומלץ להיעזר ב-`ToString`).

### הנחיות נוספות

- בעת פתרון התרגיל ניתן ואף מומלץ להגדיר מחלקות נוספות על מנת לחסוך בשכפול קוד כמה שניתן.
- בעת פתרון התרגיל, יש להקפיד על שמות משמעותיים למשתנים, לפעולות ולמחלקות.
- בעת פתרון התרגיל, יש להקפיד על הרשאות הגישה השונות בהתאם לנלמד בקורס.
- יש ליצור כל מחלקה בקובץ נפרד.
- בכל דריסה ומימוש של פעולה יש להשתמש באנוטציה `@Override` על מנת לוודא שאכן מתבצעת דריסה.
- יש להשתמש ב-`Covariant Return Type` במקומות המתאימים.
- בעת פתרון התרגיל ניתן ואף מומלץ להגדיר קבועים ולא להשתמש במספרי קסם.
- אין לייבא פעולות, מחלקות או ממשקים שלא יובאו במחלקת האיטרטור. ניתן לייבא את `java.lang.reflect.InvocationTargetException` וניתן להשתמש בחריגות אשר נובעות מהשימוש בפעולות `invoke` ו-`getMethod`.
- בכל מחלקה יש לכלול את התכונות המתאימות לה, ולספק בעבורן פעולות `get` ו-`set` **במידת הצורך בלבד**.
- על המחלקות להכיל בנאים בהתאם לצורך. אין חובה שבנאי יקבל כפרמטרים את כל התכונות אשר מוגדרות במחלקה (ואף זה אינו רצוי בעבור חלק מן המחלקות).
- בכל מחלקת חריגה אותה אתם מממשים יש להגדיר את כל **שלושת** הבנאים הסטנדרטיים עליהם דיברנו בתרגולים.
- יש לתעד את כל הפעולות והמחלקות אותן אתם מגדירים בעזרת שימוש ב-`JavaDoc` בהתאם לקובץ מוסכמות התייעוד אשר מופיע באתר הקורס. בנוסף, יש לתעד שורות קוד אשר עשויות להיות קשות להבנה.

### הרצת התוכנית וביצוע בדיקות

הועלה קובץ `Main` וקובץ פלט שמדגים שימוש במבנה הנתונים באופן כללי, וכן שימוש בשאר המחלקות.



הטכניון – מכון טכנולוגי לישראל  
הפקולטה למדעי הנדסה והחלשות  
הנדסת תוכנה אביב תשפ"ה



שימו לב, אתם תיבדקו גם אל מול טסטים אשר לא יפורסמו לכם.

**בהצלחה!**