



הנדסת תוכנה – תרגיל בית 4

דגשים להגשת המטלה

1. **תאריך הגשה: יום חמישי 28.03.2024, 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. יש להשתמש בגרסה 9.0.4 של ג'אווה בעת פתרון התרגיל.
4. מומלץ להשתמש ב-Git במהלך כתיבת התרגיל.
5. 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 שלכם (שימו לב, java. הוא סוג הקובץ).
2. אין להגיש את הקבצים כחלק מ-package.
3. הגשה אלקטרונית בלבד דרך אתר הקורס ב-moodle. ההגשה תכלול את קובץ ה- zip בלבד.
4. ההגשה מתבצעת על ידי אחד מבני הזוג לאחר שיצר קבוצה ובן הזוג השני הצטרף אליה.



הטכניון – מכון טכנולוגי לישראל
הפקולטה למדעי הנדסה והחלטות
הנדסת תוכנה חורף תשפ"ג – תשפ"ד



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.



משחק ההרפתקה הטקסטואלי

תיאור המשחק הוא בדיוק כמו בתרגיל בית 3, עם תוספות מסוימות.

הוחלט לשנות את מבנה הנתונים המשמש לחיבור החדרים ל-QuartlyLinkedList (ראו הסבר בהמשך) וכן לאפשר את העתקת המשחק כולו באמצעות העתקה עמוקה. המחלקות, הפעולות וההדפסות לא ישתנו אלא אם נאמר אחרת, אך ייתכן שעל מנת לשמור על הפונקציונליות של התרגילים הקודמים יהיה צורך בשינוי המימושים, וכן באחריות של המחלקות השונות. לשם כך תדרשו לממש מספר מחלקות **חדשות**:

מחלקת QuartNode

בתרגילים ובהרצאות ראינו את מימוש המחלקה הגנרית Node (חוליה). בתרגיל זה תממשו הרחבה למחלקה זו אשר נקראת QuartNode ומממשת את Cloneable. יש לדאוג כי QuartNode תהיה גנרית כאשר הטיפוס הגנרי מוגבל מלמעלה על ידי הממשק Cloneable. תכונות המחלקה:

- value – ערך החוליה (כאשר הטיפוס הוא הטיפוס הגנרי).
- north/east/south/west – תכונות מטיפוס המחלקה QuartNode.

פעולות המחלקה:

- פעולת set ופעולת get ל-value.
- בנאי אשר מקבל משתנה מהטיפוס הגנרי ומאתחל את שאר התכונות ל-null.
- בנאי אשר מקבל משתנה מהטיפוס הגנרי, Direction וחוליה other ויוצרת חוליה חדשה שתהיה מחוברת ל-other מהכיוון שהתקבל, ובהתאם other תהיה מחוברת בכיוון ההופכי לחוליה הנוצרת.
- פעולת getNeighbor אשר מקבלת Direction ומחזירה את החוליה המחוברת בכיוון זה.
- פעולת clone אשר תדאג להפעלת clone על value. העתקת השכנים תתבצע במחלקה אחרת.

מחלקת QuartlyLinkedList

מחלקה זו תהיה הכללה של LinkedList אשר בה נתקלנו בתרגילים ובהרצאות. מחלקה זו תהיה גנרית כאשר הטיפוס הגנרי מוגבל מלמעלה על ידי הממשק Cloneable. בנוסף, המחלקה תממש את Iterable ואת Cloneable.

תכונות המחלקה:

- root – עצם מטיפוס QuartNode אשר ייצג את התחלת הרשימה.

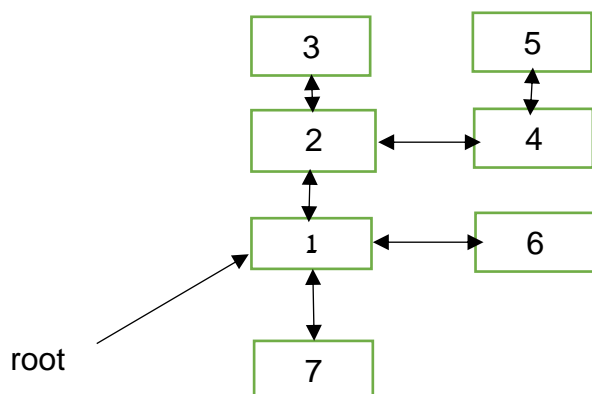
פעולות המחלקה:

- בנאי אשר יאתחל את root ל-null.

- getter למשתנה המחלקה.
- פעולת add אשר תקבל משתנה גנרי חדש toInsert להכנסה לרשימה, משתנה גנרי target אשר קיים ברשימה (יש צורך לחפש אותו וייתכן כי לא יהיה קיים) ו-Direction שמייצג את הכיוון אליו יתחבר האיבר החדש לאיבר הקיים. הפעולה תוסיף QuartNode חדש שערכו יהיה toInsert ויחובר לחוליה המתאימה שמכילה את target (תוכלו להניח כי קיימת אחת כזאת לכל היותר) מכיוון Direction. אם המשתנה target לא קיים ברשימה, יש לזרוק חריגה בלתי מסומנת בשם NoSuchElementException. אם לא ניתן לחבר חוליה חדשה מכיוון זה (עקב כך שהכיוון תפוס) תזרוק חריגה בלתי מסומנת DirectionIsOccupied. שימו לב, אם הרשימה ריקה אין צורך לבדוק את target ואת Direction ומיד ליצור חוליה שתכיל את toInsert אשר תהיה root.
- פעולת remove אשר תקבל משתנה גנרי חדש toRemove ותסיר אותו מהרשימה. אין צורך לדאוג לחיבור השכנים גם אם נותקו לחלוטין מן השאר. אם toRemove לא נמצא ברשימה תזרוק חריגה בלתי מסומנת NoSuchElementException.
- פעולת clone אשר תדאג להעתקה עמוקה של הרשימה.

מחלקת QuartlyLinkedListIterator

המחלקה מייצגת את האיטרטור על עצמים מטיפוס QuartlyLinkedList, וכמובן תממש את הממשק Iterator. המעבר על הרשימה יתבצע באמצעות אלגוריתם DFS כאשר תחילת ההדפסה היא מ- root. אלגוריתם זה משמש לחיפוש, כאשר מכל חוליה מחפשים את כלל החוליות אליהן ניתן להגיע מהחיבור הצפוני שלה, ולאחר מכן המזרחי, הדרומי ולבסוף המערבי. כלומר בהינתן חוליה ויציאה מחפשים לעומק כל את החוליות הנגישות מיציאה זו. בנוסף, על כל חוליה עוברים פעם אחת ויחידה. לדוגמה, עבור QuartlyLinkedList הבאה:



תוצאת הרצת DFS מ- root תעבור על החוליות בסדר הבא (משמאל לימין) 1,2,3,4,5,6,7.

אם הרשימה ריקה, האיטרטור יכיל null בתור הערך הבא עם אתחולו. שימו לב, האיטרטור יחזיר איטרטיבית את החוליות לפי סדר המעבר עליהן ב-DFS, כאשר חוליה אשר הוזכרה בעבר לא תוזכר שוב. לשימושכם מצורף קוד המחלקה.



השינויים במימוש המשחק:

- בהוספת חדר אין צורך בהדפסת הודעת שגיאה.
- בהסרת חדר אין צורך בהדפסת הודעת שגיאה וכן תוכלו להניח כי לא יוסר חדר אשר השחקן נמצא בו.
- בעת חיבור חדרים אין צורך להדפיס הודעת שגיאה.
- הוספת פעולות clone למחלקות השונות (שימו לב, עליהן להיות public).

השינויים במחלקות הישנות:

מחלקת GameManager

- כעת החדרים ישמרו בעזרת QuartlyLinkedList, ולא במערך.
- מספר החדרים לא חסום.
- לא תהיה פעולת connectRooms ופעולת addRoom תקבל חדר להוספה למשחק toInsert, חדר target אשר קיים במשחק (ויש צורך לבדוק זאת) וכיוון חיבור Direction direction אשר מייצג את היציאה של target אליו יתחבר החדר החדש. אם הפעולה הצליחה יש להדפיס toInsert was added and is connected to target from the direction exit. אם הפעולה הכניסה חדר לרשימה ריקה, יש להדפיס toInsert was added. אחרת, אם target לא קיים יש לזרוק חריגה בלתי מסומנת RoomDoesNotExist. אם לא ניתן לחבר חדר חדש מכיוון זה (עקב כך שהכיוון תפוס) תזרוק חריגה בלתי מסומנת ExitIsOccupied.
- פעולת removeRoom לא תדפיס הודעה כאשר לא קיים חדר השווה לחדר המתקבל כפרמטר, אלא תזרוק חריגה בלתי מסומנת RoomDoesNotExist.
- הוספת פעולת clone אשר תבצע העתקה עמוקה של כל המשחק. **הכוונה:** תוכלו להשתמש בפעולות getMethod ו-invoke לצורך העתקת המשחק.

מחלקת Room

שימו לב כי כעת התכונות של Room ישתנו.

לצורכי בדיקה אנא הוסיפו את הפעולה toString אשר תחזיר מחרוזת עם המבנה: "Room: <name>", כאשר <name> מייצג את שם החדר (לא להדפיס את הסוגריים המשולשים).

כלל המחלקות הישנות



כלל המחלקות הישנות צריכות לממש את Cloneable. זכרו להשתמש ב-Covariant Return Type במימוש clone.

הנחיות מקלות

כלל ההנחיות של התרגילים הקודמים.

הנחיה לפתרון

שימו לב כי ניתן להשתמש באיטרטור בכל הפעולות אשר מקבלות כפרמטר חדר על מנת לחפש את אותו החדר. את מימוש הclone תוכלו לבצע בעזרת מעבר איטרטיבי על החוליות עם האיטרטור ושמירת העבר של החוליות שנוצרו באמצעות העתקה - בעזרת מחסנית - תוך כדי שמירת החוליות שכבר הועתקו ברשימה לוקלית כלשהי.

הנחיות נוספות

- בעת פתרון התרגיל ניתן ואף מומלץ להגדיר מחלקות נוספות על מנת לחסוך בשכפול קוד כמה שניתן. עם זאת, אין להוסיף פעולות אשר משתמשות במחלקות אותן הגדרתן בתרגיל הקודם (לא בצורה ישירה ולא בצורה עקיפה) למחלקות מבנה הנתונים הגנרי.
- 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.
- 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

בהמשך יועלה קובץ Main עם מספר טסטים. כאשר יעלה קובץ Main יעלה גם קובץ HW4_output.txt על מנת שתוכלו לבצע השוואה עם קובץ הפלט שלכם. שימו לב, אתם תיבדקו גם אל מול טסטים אשר לא יפורסמו לכם.

בהצלחה!