

עבודת בית 3

תכנות מונחה עצמים

מחלקות אבסטרקטיות, ממשקים וגנריות

מחלקות אבסטרקטיות, שיטות אבסטרקטיות, דריסה. יחס "is a".

ממשקים. שיטות ממשקיות. קבועים ממשקיים ושיטות סטטיות ממשקיות.

מילת המפתח extends. מילת המפתח implements. מילת המפתח abstract. מילת המפתח interface.

שיטות גנריות ומחלקות גנריות.

המחלקה למדעי המחשב

המכללה האקדמית להנדסה ע"ש סמי שמעון

הנחיות

מועד פרוסום: 15.07.2024 מועד אחרון להגשה: 04.08.2024

מתרגלת אחראית: יעל וקסלר

מועד הגשה

1. ההגשה היא עד סוף יום ההגשה, כלומר עד השעה 23:59 ביום המצוין לעיל כמועד אחרון להגשה. אל תחכו לרגע האחרון. תכננו את זמנכם בהתאם. הגישו לפני.
2. איחור במועד ההגשה יגרור הורדה של ציון, 5 נק' לכל יום איחור או חלק ממנו. בכל מקרה לא יהיה ניתן להגיש מעבר ל-2 ימי איחור ממועד ההגשה דלעיל.

אופן הגשה

3. קראו היטב את השאלות. עליכם לענות על כל השאלות בעבודה זו.
4. הגשת העבודה תהיה דרך אתר הקורס במודל בלבד. הגשת העבודה היא ביחידים.
5. כיצד להגיש?
- a. כל הקבצים המוגשים יהיו קבצי java, כלומר עם סיומת java. כל מחלקה צריכה לשבת בקובץ משלה עם אותו שם כמו שם המחלקה. כל הקבצים הקשורים למשימה 1 ישבו בספריה (package, חבילה) בשם queue ומשימה 2 ישבו בספריה (package, חבילה) בשם school.
- b. יש להגיש רק תוכנית שעוברת קומפילציה תקינה ללא שגיאות על קומפיילר שפת java.
- c. כיצד להעלות הקבצים למודל. למודל יש להעלות קובץ אחד:
יש ליצור תיקייה ראשית. בתוכה:
 - i. תת תיקייה queue ובתוכה המחלקה ArrayQueue שקשורה למשימה 1.
 - ii. תת תיקייה school ובתוכה כל המחלקות שקשורות למשימה 2.
 - iii. תת תיקייה set ובתוכה כל המחלקות שקשורות למשימה 3.(יש לשים לב ששם הספרייה שבה שומרים את המחלקות, יהיה תואם לשם החבילה (package) שבה הגדרתם את המחלקות).
את התיקייה הראשית יש לכווץ ולהעלות למודל.
6. בתחילת כל קובץ יש להוסיף את התיעוד הבא. יש לשנות את השם לשם שלכם ואת תעודת הזהות לתעודת הזו שלכם.
- // Assignment: 3
// Author: Israel Israeli, ID: 01234567
7. לאחר שהעליתם את הקובץ שלכם למודל, הורידו אותם מהמודל למחשב שלכם וודאו כי הקובץ תקין וכי העליתם את הקובץ הנכון והמלא. לאחר תום מועד ההגשה לא

יתקבלו ערעורים על כך שהעליתם קבצים לא תקינים או שהעליתם בטעות קבצים אחרים / לא נכונים.

שאלות

8. שאלות בנוגע העבודה יש לשאול **בפורום באתר המודל של הקורס** או בשעות קבלה של המתרגל/ת האחראי/ת בלבד. אין לשלוח שאלות במייל לא למתרגל/ת האחראי/ת ולא למתרגלים/מרצים אחרים.
9. ניתן לשאול שאלות הבהרה ומיקוד על המשימות שבעבודה במידה ומשימה מסוימת לא ברורה. לא ניתן לשאול על הפתרונות שלכם. לדוגמא, לא ניתן לשאול האם הפתרון שלי נכון, לא ניתן לשאול למה הפתרון לא עובד, וכדומה.

קוד

10. ניתן להשתמש בכל החומר שלמדנו עד למועד פרסום העבודה. בפרט, החומר עוסק וכולל את כל החומר שלמדנו עד לממשקים – כולל ממשקים. אך לא כולל מחלקות גנריות ולא כולל מחלקות מיוחדות.
11. הדבר החשוב ביותר הוא שהקוד שלכם יעבוד באופן תקין / נכון. כמו כן, חשוב שהקוד יהיה בנוי בצורה התואמת את פרדיגמת התכנות מונחה העצמים כפי שלמדנו עד כה.
12. יש להקפיד על תכנות נכון. יש לתעד את הקוד באופן מלא בפורמט Javadoc בלבד ובאנגלית בלבד. יש להקפיד על הזחות, שמות משמעותיים וכיוצא באלו.

שונות

13. המשימות בעבודה זו הינן שוות משקל. כלומר, משקל כל משימה הוא 100 חלקי מספר המשימות בעבודה.
14. אם לא צוין במפורש אחרת במשימה ספציפית, בכל המשימות בעבודה זו הניחו כי הקלט מקיים את כל ההנחות הכתובות במשימה. כלומר, אינכם נדרשים לבדוק שהקלט מקיים את ההנחות, אלא אם כן צוין במפורש במשימה שצריך.
15. באפשרותכם להשתמש בשיטות של הספרייה הסטנדרטית במידה ואינן פותרות את המשימה כולה/רובה או חלק גדול ממנה. במידה ואתם בספק, ניתן להעלות את השאלה בפורום העבודות.

בהצלחה!

עבודת בית 3

מחלקות אבסטרקטיות, ממשקים וגנריות

משימה 1: ממשקים – מימוש תור (queue)

תזכורת: כל המחלקות במשימה זו צריך לשמור בספריה queue.

במדעי המחשב, תור (queue) הוא מבנה נתונים מופשט המוגבל בגודלו ומייצג אוסף נתונים המסודר על פי סדר מוגדר.

התור מוגדר על ידי הפעולות הבאות:

- הכנסה (enqueue) - הוספת אובייקט אחד חדש בסופו של התור.
- הוצאה (dequeue) - הוצאתו של האובייקט הנמצא בראש התור.
- בדיקת מהו הערך בראש התור (peek).
- בדיקת מהו מספר האובייקטים בתור (size).
- בדיקה האם התור ריק (isEmpty).
- בדיקה האם התור מלא (isFull).

כל הפעולות מתבצעות בסיבוכיות $O(1)$ מלבד dequeue שמתבצעת בסיבוכיות $O(\text{size})$. פעולות ההכנסה וההוצאה בתור מבוססות על העיקרון נכנס ראשון - יוצא ראשון (first in first out) FIFO. השלימו את המחלקה ArrayQueue המממשת את הממשק Queue המצורף למטלה. המחלקה מכילה שני שדות בלבד **אסור** להוסיף או להוריד שדות מחלקה. בנו בנוי עבור המחלקה, המקבל את הפרמטר capacity המייצג את קיבולת התור. הבנאי צריך לאתחל את השדות במחלקה כאשר:

- השדה queue מייצג מערך אובייקטים (מסוג object) שקיימים בתור. ביצירה התור ריק מאובייקטים.
 - השדה size מייצג את כמות האובייקטים שקיימים בתור. ביצירת התור, הוא ריק. כלומר, מספר האובייקטים בתור הוא אפס.
- המחלקה תממש את הממשק על פי ההנחיות הבאות:

❖ **הכנסה (enqueue):**

מוסיף אלמנט לסוף התור, מחזיר true אם האלמנט נוסף בהצלחה אחרת מחזיר false. שימו לב לא ניתן להוסיף אובייקט לתור אם התור מלא. כלומר, אם הקיבולת של התור הוא 20 וגודלו (מספר האובייקטים בתוכו) הוא 20, אז כאשר מנסים להכניס אובייקט נוסף לתור השיטה מחזירה false מפני שהתור מלא ולא ניתן להכניס אובייקטים נוספים לתור.

❖ **הוצאה (dequeue) - הוצאתו של האובייקט הנמצא בראש התור:**

מסיר אלמנט מראש התור (במקום האפס של המערך) ומחזיר אותו, אם התור ריק יש להחזיר null אחרת יש לעדכן את השדות queue ו-size בהתאם כך שהאיבר הראשון במערך יהיה האיבר הראשון כעת בתור. את המערך queue נעדכן ע"י הזזה שמאלה של האובייקטים שבתור זאת מאחר שהמקום האפס בתור התפנה, ראש התור התפנה ובנוסף יש להקטין את הגודל של התור באחת כלומר את השדה size.

❖ **בדיקת מהו הערך בראש התור (peek):**

מחזיר את האלמנט שנמצא בראש התור מבלי להסיר אותו, אם התור הוא ריק מחזירה null. שימו לב התור לא משתנה.

❖ **בדיקת מהו מספר האובייקטים בתור (size):**

מחזיר את מספר האלמנטים שקיימים כעת בתור, אם התור ריק מחזיר 0.

❖ בדיקה האם התור ריק (isEmpty):

מחזיר true אם התור ריק, אחרת מחזיר false.

❖ בדיקה האם התור מלא (isFull):

מחזיר true אם התור מלא, אחרת מחזיר false. תור יחשב מלא אם מערך האובייקטים מלא. כלומר מספר האובייקטים במערך שווה לאורך המערך.

כתבו main המציג את עבודתכם על ידי יצירת אובייקטים שהגדרתם והפעלת כל המתודות (סדר ההפעלה לבחירתכם).

דוגמא:

המערך שמצויר הוא השדה queue של המחלקה והתא השמאלי ביותר הוא אפס (האינדקס שלו 0) התא מימינו הוא התא האחד וכך הלאה. התור בדוגמא בעל קיבולת 6, ביצירתו התור ריק

--	--	--	--	--	--

לאחר פעולת enqueue(7)

7					
---	--	--	--	--	--

לאחר פעולת enqueue(12)

7	12				
---	----	--	--	--	--

לאחר פעולת enqueue(8)

7	12	8			
---	----	---	--	--	--

לאחר פעולת enqueue(102)

7	12	8	102		
---	----	---	-----	--	--

לאחר פעולת enqueue(12)

7	12	8	102	12	
---	----	---	-----	----	--

לאחר פעולת dequeue(), מוחזר הערך 7

12	8	102	12		
----	---	-----	----	--	--

פעולת peek(), תחזיר את הערך 12

12	8	102	12		
----	---	-----	----	--	--

לאחר פעולת dequeue(), מוחזר הערך 12

8	102	12			
---	-----	----	--	--	--

פעולת size(), תחזיר את הערך 3

8	102	12			
---	-----	----	--	--	--

פעולת isFull(), תחזיר את הערך false

8	102	12			
---	-----	----	--	--	--

פעולת isEmpty(), תחזיר את הערך false

8	102	12			
---	-----	----	--	--	--

משימה 2 : מחלקות אבסטרקטיות – בית ספר (school)

תזכורת: כל המחלקות במשימה זו צריך לשמור בספריה school.

בתרגיל זה אתם מתבקשים לבנות תוכנה המדמה מערכת עובדים בית ספרית. בית הספר מורכב מסוגים שונים של עובדים (Employee), אנו נחלק את העובדים ל-3 קבוצות הנהלה (Administrator), הוראה (Teacher) וסגל כללי (SupportStaff).

במחלקת ההנהלה (Administrator) קיימים שני סוגי עובדים והם מנהל (Principal) וסגן מנהל (VicePrincipal), במחלקת ההוראה (Teacher) קיימים שלושה סוגי מורים והם מורה למתמטיקה (MathTeacher), מורה לאנגלית (EnglishTeacher) ומורה למדעים (ScienceTeacher), במחלקת הסגל הכללי (SupportStaff) קיימים שני סוגי עובדים והם שרת (Janitor) ומזכירה (Secretary).

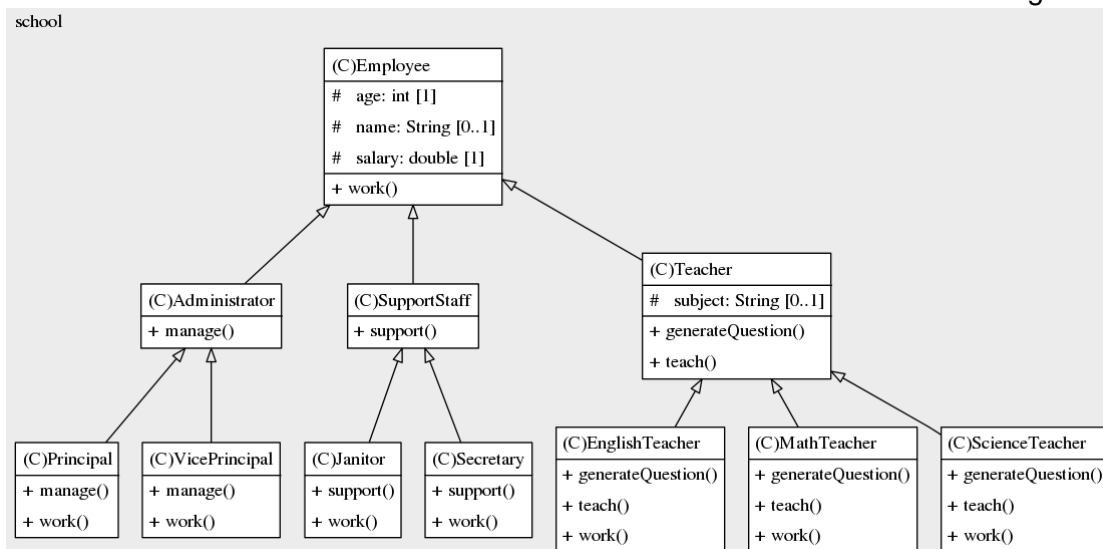
- המחלקות SupportStaff, Administrator, Teacher, Employee הן מחלקות אבסטרקטיות.
- המחלקות SupportStaff, Administrator, Teacher יורשות מהמחלקה Employee.
- המחלקות VicePrincipal, Principal יורשות מהמחלקה Administrator.
- המחלקות MathTeacher, EnglishTeacher, ScienceTeacher יורשות מהמחלקה Teacher.
- המחלקות Secretary, Janitor יורשות מהמחלקה SupportStaff.

כל המחלקות צריכות לכלול בנאי, השיטות toString ו-equals. אתם יכולים להוסיף שדות או שיטות כרצונכם.

לפניכם דיאגרמת מחלקות המתארת את המחלקות, את ההיררכיה ביניהן, את השדות והמתודות הקיימים בכל אחת מהן, הקפידו על התיאור המילולי של כל מתודה וכל שדה. הדיאגרמה אינה מראה את כל השיטות, לדוגמא, בנאים, שיטות get/set, וכדומה. היא מראה את השיטות העיקריות/מרכזיות. **התעלמו מהמספרים ליד השדות**

לפני מסמן כי יש להגדיר protected.
+ לפני מסמן כי יש להגדיר public.

כאשר הגישה לשדות היא מחוץ למחלקה, כלומר על ידי מחלקה אחרת היא תתבצע בעזרת get/set בלבד.



המחלקה האבסטרקטית Employee מכילה את השדות

- name – מחרוזת המייצגת את השם המלא של העובד
- age - מספר שלם המייצג את גיל העובד
- salary – מספר ממשי המייצג את

המחלקה מממשת את set ו get לכל אחד מן השדות.
הבנאי מתאים ליותרת הבאה:

```
public Employee(String name, int age, double salary)
```

המחלקה מכילה בנוסף מתודה אבסטרקטית work המחזירה void.

המחלקה האבסטרקטית Administrator יורשת מהמחלקה Employee
הבנאי מתאים ליותרת הבאה:

```
public Administrator (String name, int age, double salary)
```

המחלקה מכילה בנוסף מתודה אבסטרקטית manage המחזירה void.

המחלקה האבסטרקטית Teacher יורשת מהמחלקה Employee מכילה את השדה

- subject – מחרוזת final המייצגת את המקצוע שהמורה מלמד.

המחלקה מממשת את get לשדה.
הבנאי מתאים ליותרת הבאה:

```
public Teacher(String name, int age, double salary, String subject)
```

המחלקה מכילה בנוסף מתודה אבסטרקטית teach המחזירה void. ומתודה אבסטרקטית
נוספת generateQuestion המחזירה void.

המחלקה האבסטרקטית SupportStaff יורשת מהמחלקה Employee
הבנאי מתאים ליותרת הבאה:

```
public SupportStaff (String name, int age, double salary)
```

המחלקה מכילה בנוסף מתודה אבסטרקטית support המחזירה void.

המחלקה VicePrincipal יורשת מהמחלקה Administrator
הבנאי מתאים ליותרת הבאה:

```
public VicePrincipal (String name, int age, double salary)
```

המחלקה מממשת את המתודות work וmanage (override) כך שהמתודה work תדפיס:
"The vice principal is working".
והמתודה manage תדפיס:
"The vice principal is assisting the principal."

המחלקה Principal יורשת מהמחלקה Administrator
הבנאי מתאים ליותרת הבאה:

```
public Principal (String name, int age, double salary)
```

המחלקה מממשת את המתודות work וmanage (override) כך שהמתודה work תדפיס:
"The principal is working. "
והמתודה manage תדפיס:
" The principal is managing the school."

המחלקה ScienceTeacher יורשת מהמחלקה Teacher
הבנאי מתאים לכותרת הבאה ומאתחל את subject בערך Science:
public ScienceTeacher (String name, int age, double salary)
המחלקה מממשת את המתודות work, teach, generateQuestion (override) כך שהמתודה work תדפיס:

"The science teacher is working."

המתודה teach תדפיס:

"The science teacher is teaching science."

המתודה generateQuestion יוצרת שאלת פיזיקה הכוללת חישוב כוח באמצעות מסה ותאוצה, היא תיצור מסה אקראית בין 0 ל-100 כולל קילוגרם, ותאוצה אקראית בין 0 ל-10 כולל מטר לשנייה בריבוע

ותחשב את התוצאה הנכונה של הכוח המתקבל ביחידות מידה ניוטון באמצעות הנוסחה $F = m \cdot a$ כלומר מסה כפול תאוצה. היא תדפיס למסך את השאלה ולאחר מכן תקלוט מהמשתמש את תשובתו ותבדוק אם התשובה נכונה או לא ותדפיס הודעה בהתאם. כלומר אם התשובה נכונה תדפיס "כל הכבוד, תשובה נכונה. אחרת תדפיס, "התשובה הנכונה היא" ואת התשובה הנכונה.

המחלקה MathTeacher יורשת מהמחלקה Teacher
הבנאי מתאים לכותרת הבאה ומאתחל את subject בערך Mathematics :
public MathTeacher (String name, int age, double salary)
המחלקה מממשת את המתודות work, teach, generateQuestion (override) כך שהמתודה work תדפיס:

"The math teacher is working."

המתודה teach תדפיס:

"The math teacher is teaching math. "

המתודה generateQuestion יוצרת שאלת מתמטיקה מתוך לוח הכפל כלומר חישוב מכפלה של שני מספרים אקראיים בין 0 ל-10 כולל ותחשב את התוצאה הנכונה. היא תדפיס למסך את השאלה ולאחר מכן תקלוט מהמשתמש את תשובתו ותבדוק אם התשובה נכונה או לא ותדפיס הודעה בהתאם. כלומר אם התשובה נכונה תדפיס "כל הכבוד, תשובה נכונה. אחרת תדפיס, "התשובה הנכונה היא" ואת התשובה הנכונה.

המחלקה EnglishTeacher יורשת מהמחלקה Teacher
הבנאי מתאים לכותרת הבאה ומאתחל את subject בערך English:
public EnglishTeacher (String name, int age, double salary)
המחלקה מממשת את המתודות work, teach, generateQuestion (override) כך שהמתודה work תדפיס:

"The english teacher is working."

המתודה teach תדפיס:

"The english teacher is teaching english."

המתודה generateQuestion יוצרת שאלת על ידי בחירה אקראית של מילה מתוך רשימה קבועה והיא

{"cat", "dog", "tree", "book", "cup", "sun", "moon", "star", "car", "house"}

ועל ידי בחירה אקראית של הנחיה מתוך רשימה קבועה והיא

{"Spell", "Define", "Use in a sentence", "Rhyme with", "Antonym of", "Synonym of"}

היא תדפיס למסך את ההנחיה ואת המילה ולאחר מכן תקלוט מהמשתמש את תשובתו כמחרוזת ותדפיס אותה למסך.

המחלקה Janitor יורשת מהמחלקה SupportStaff

הבנאי מתאים ליותרת הבאה:

public Janitor (String name, int age, double salary)

המחלקה מממשת את המתודות work וsupport (override) כך שהמתודה work תדפיס:

"The janitor is working ."

והמתודה support תדפיס רשימת הזמנות אקראית כך שבכל שורה יופיעו שם פריט וכמות (אין

צורך ליצור שמות ממשיים אלא "Item #1", "Item #2" וכן הלאה)

מספר השורות יהיה 10 כלומר 10 פריטים ברשימה והכמויות ינועו בין 0 ל-10 בלבד.

המחלקה Secretary יורשת מהמחלקה SupportStaff

הבנאי מתאים ליותרת הבאה:

public Secretary (String name, int age, double salary)

המחלקה מממשת את המתודות work וsupport (override) כך שהמתודה work תדפיס:

"The secretary is working ."

והמתודה support תדפיס טבלת ציונים אקראית כך שבכל שורה יופיעו שם סטודנט וציון

סטודנט (אין צורך ליצור שמות ממשיים אלא "Student #1", "Student #2" וכן הלאה)

מספר השורות יהיה 30 כלומר 30 סטודנטים והציונים בין 0 ל-100 בלבד.

התוכנית הראשית SchoolSystem מצורפת למטלה. פלט לדוגמא

The principal is working.

The principal is managing the school.

The vice principal is working.

The vice principal is assisting the principal.

The math teacher is working.

The math teacher is teaching math.

What is 9 * 0?

Please enter your answer :

0

Well done your answer is correct

The English teacher is working.

The English teacher is teaching English.

Antonym of the word 'book.'

Please enter your answer :

unbook

Your answer is: unbook

The science teacher is working.

The science teacher is teaching science.

A 17.37496038890486 gram object is accelerating at 5.03910900394372 m/s^2.

What is the resulting force in Newtons?

Please enter your answer :

100

Your answer is incorrect. The correct answer is: 87.55431933889595

The janitor is working.

The janitor is ordering the following items:

Item #1 amount:6

Item #2 amount:9

Item #3 amount:6

Item #4 amount:0

Item #5 amount:4

Item #6 amount:10

Item #7 amount:2

Item #8 amount:3

Item #9 amount:5

Item #10 amount:3

The secretary is working.

The secretary enters the following grades into moodle:

Student #1	grade:23
Student #2	grade:42
Student #3	grade:19
Student #4	grade:100
Student #5	grade:62
Student #6	grade:75
Student #7	grade:2
Student #8	grade:15
Student #9	grade:68
Student #10	grade:43
Student #11	grade:74
Student #12	grade:7
Student #13	grade:94
Student #14	grade:8
Student #15	grade:40
Student #16	grade:19
Student #17	grade:13
Student #18	grade:18
Student #19	grade:13
Student #20	grade:69
Student #21	grade:89
Student #22	grade:27
Student #23	grade:17
Student #24	grade:68
Student #25	grade:56
Student #26	grade:5
Student #27	grade:98
Student #28	grade:60
Student #29	grade:80
Student #30	grade:77

משימה 3: מחלקות גנריות – קבוצה (Set)

תזכורת: את כל המחלקות של משימה זו צריך לשמור בספריה set.

קבוצה (Set) היא סוג של מבנה נתונים מופשט שכל ערך מופיע בו לכל היותר פעם אחת (אין כפילויות), ואין חשיבות לסדר בין הערכים. מימוש של קבוצה הוא למעשה ייצוג ממוחשב של קבוצה מתמטית סופית.

כתבו ממשק גנרי Set המכיל את השיטות שניתן לבצע על הקבוצה כך שהאלמנטים שבה הן מסוג `<E>`. על הקבוצה ניתן יהיה לבצע את הפעולות הבאות:

- הוספת איבר (add) - הוספת אובייקט אם אינו נמצא בקבוצה.
- הוצאת איבר (remove) - הוצאתו של אובייקט מסוים מהקבוצה.
- חיפוש ערך (contains) - בדיקה אם אובייקט מסוים שייך לקבוצה.
- בדיקת מהו מספר האובייקטים בקבוצה (size).
- בדיקה האם הקבוצה ריקה (isEmpty).

כתבו את המחלקה הגנרית `SimpleSet` המממשת את הממשק הגנרי `Set`.

בנו בנאי עבור המחלקה, המקבל את הפרמטר `capacity` המייצג את הקיבולת ההתחלתית הקבוצה. הבנאי צריך לאתחל את השדות במחלקה כאשר:

- השדה `elements` מייצג מערך בגודל `capacity` המכיל אובייקטים שקיימים בקבוצה. ביצירה הקבוצה ריקה.
- השדה `size` מייצג את כמות האובייקטים שקיימים כעת בקבוצה. ביצירה הקבוצה ריקה כלומר, מספר האובייקטים הוא אפס.
- השדה `capacity` מייצג את כמות האובייקטים המקסימלית שקבוצה יכולה להכיל. מתעדכן ויוצר מערך חדש במידה וישנו ניסיון להוסיף אובייקט כשמגיע למקסימום. כלומר, כש `size` - `capacity` שווים זה לזה.

בנו בנאי ברירת מחדל עבור המחלקה, המאתחל את הפרמטר `capacity` בערך 5, הפרמטר `capacity` מייצג את קיבולת הקבוצה. בנוסף הבנאי יאתחל את השדות במחלקה באופן דומה לבנאי (שאינו ברירת מחדל).

המחלקה תממש את המתודות על פי ההנחיות הבאות:

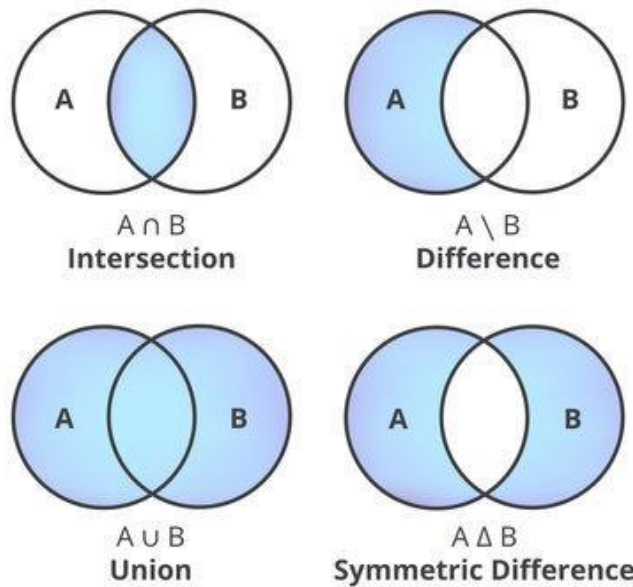
- הוספת איבר (add):
הוספת אובייקט אם אינו נמצא בקבוצה. המתודה תקבל פרמטר e שהינו אלמנט חדש שנרצה להוסיף לקבוצה, היא תעבור על כל האלמנטים בקבוצה ואם e אינו מוכל בה תוסיף אותו. המתודה תבדוק שאיננו חורגים מה (capacity כלומר שהsize קטן מהcapacity). אם size גדול או שווה ל-capacity, נגדיל את הקיבולת ב-5, נשים לב לעדכן את השדות capacity ו-elements בהתאם (נעתיק את המערך הישן וניצור מערך חדש גדול יותר המכיל את כל האלמנטים של המערך הישן והאלמנט החדש e).
- הוצאת איבר (remove):
הוצאתו של אובייקט מסוים מהקבוצה. המתודה תקבל פרמטר e שהינו אלמנט שנרצה להסיר מהקבוצה, מחזיר true אם האלמנט קיים בקבוצה ומעדכנת את השדות size ו-elements בהתאם כלומר, מקטינה את size ב-1 ומעדכנת את המערך ע"י שמירה על רציפות (נעביר את האובייקט שנרצה להסיר לסוף המערך ונשנה את הערך ל-null). אם האלמנט לא קיים בקבוצה תחזיר false ולא תעדכן אף שדה.
שימו לב המערך צריך לשמור על רציפות.
למשל אם הקבוצה היא {1,2,13,4,9} ובחרנו להסיר את 2 אז המערך צריך להיות {1,13,4,9} אם גודל המערך הוא 5 וגודל הקבוצה הוא 4 אז האיבר באינדקס 4 צריך להיות null.
- חיפוש ערך (contains):
בדיקה אם אובייקט מסוים שייך לקבוצה. המתודה תקבל פרמטר e שהינו אלמנט שנרצה לבדוק שייכות לקבוצה, מחזיר true אם האלמנט קיים בקבוצה, אחרת כלומר אם האלמנט לא קיים בקבוצה תחזיר false.
- בדיקת מהו מספר האובייקטים בקבוצה (size):
מחזיר את מספר האלמנטים שקיימים כעת בקבוצה, אם הקבוצה ריקה מחזיר 0.
- בדיקה האם הקבוצה ריקה (isEmpty):
מחזיר true אם הקבוצה ריקה, אחרת מחזיר false.

נגדיר מחלקה SetUtils המספקת שיטות סטטיות וגנריות לביצוע פעולות על קבוצות. ניתן יהיה לבצע את הפעולות הבאות:

- איחוד (union) – יוצר קבוצה המכילה את כל מה ששייך לקבוצה A או לקבוצה B. המתודה מקבלת שתי קבוצות כפרמטר ותחזיר את תוצאת האיחוד שלהן.
- חיתוך (intersection) – יוצר קבוצה המכילה את כל מה ששייך גם לקבוצה A וגם לקבוצה B. המתודה מקבלת שתי קבוצות כפרמטר ותחזיר את תוצאת החיתוך שלהן.
- הפרש (difference) – יוצר קבוצה המכילה את כל מה ששייך לקבוצה A ולא שייך לקבוצה B. המתודה מקבלת שתי קבוצות כפרמטר הראשון הוא A והפרמטר השני הוא B ותחזיר את תוצאת ההפרש שלהן כך ש $A \setminus B$ הקבוצה הראשונה בפרמטרים היא A והקבוצה השנייה היא B (הסדר הוא משמאל לימין).
- הפרש סימטרי (symmetricDifference) – יוצר קבוצה המכילה את כל מה ששייך גם לקבוצה A וכל מה ששייך לקבוצה B אך לא שייך לשתייהן. המתודה מקבלת שתי קבוצות כפרמטר ותחזיר את תוצאת ההפרש הסימטרי שלהן.
- קבוצת חזקה (powerSet) - יוצר קבוצה המכילה את כל תתי הקבוצות של קבוצה A, שימו לב כי קבוצת החזקה $P(A)$ כוללת, בין היתר, את הקבוצה הריקה ואת הקבוצה A עצמה. המתודה מקבלת קבוצה כפרמטר ותחזיר את קבוצת כל תתי הקבוצות על פי ההגדרה

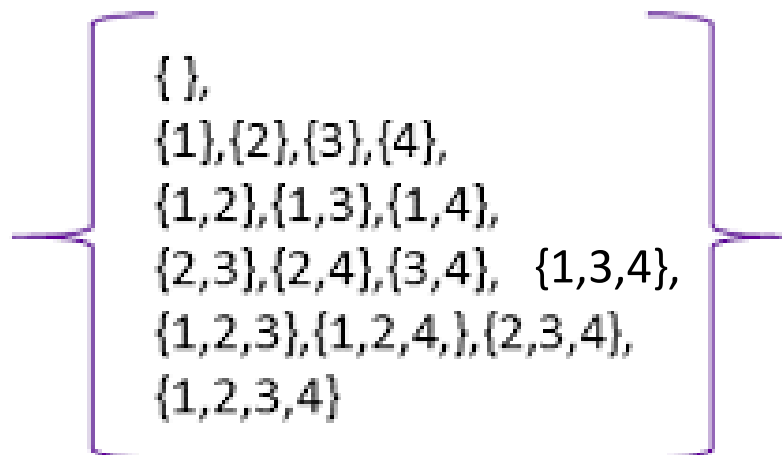
$$P(A) = \{B \mid B \subseteq A\}.$$

Sets and Venn Diagrams



הפעולות (איחוד, חיתוך, הפרש והפרש סימטרי) בין קבוצות מתוארות באיור.

Power Set of $\{1,2,3,4\}$



דוגמא לקבוצת החזקה של הקבוצה $\{1,2,3,4\}$ מתוארת באיור. שימו לב כי מספר תתי הקבוצות של קבוצה בגודל n הוא 2^n .

□ איחוד (union):

יוצר קבוצה חדשה המכילה את כל מה ששייך לקבוצה A או לקבוצה B. כלומר המתודה תקבל שני פרמטרים A ו-B שהינם מופעים של SimpleSet שנרצה לבצע איחוד ביניהם, היא תעבור על כל האלמנטים בקבוצות וכאשר ישנו אלמנט שאינו מוכל בקבוצה החדשה שיצרנו היא תוסיף אותו אל elements של הקבוצה החדשה, ניתן להשתמש במתודה add שכתבתם על מנת לעדכן את כל השדות בהתאם והחזירו את הקבוצה החדשה מהפונקציה.

□ חיתוך (intersection):

יוצר קבוצה חדשה המכילה את כל מה ששייך לקבוצה A וגם לקבוצה B. כלומר המתודה תקבל שני פרמטרים A ו-B שהינם מופעים של SimpleSet שנרצה לבצע חיתוך ביניהם, היא תעבור על כל האלמנטים בקבוצות וכאשר ישנו אלמנט שמוכל בשתי הקבוצות היא תוסיף אותו אל elements של הקבוצה החדשה, ניתן להשתמש במתודה add שכתבתם על מנת לעדכן את כל השדות בהתאם והחזירו את הקבוצה החדשה מהפונקציה.

□ הפרש (difference):

יוצר קבוצה חדשה המכילה את כל מה ששייך לקבוצה A ולא שייך לקבוצה B. כלומר המתודה תקבל שני פרמטרים A ו-B שהינם מופעים של SimpleSet שנרצה לבצע הפרש ביניהם, היא תעבור על כל האלמנטים בקבוצה A (תזכורת: הפרמטר הראשון A הוא השמאלי מביניהם) וכאשר ישנו אלמנט שמוכל בקבוצה A ולא מוכל בקבוצה B היא תוסיף אותו אל elements של הקבוצה החדשה, ניתן להשתמש במתודה add שכתבתם על מנת לעדכן את כל השדות בהתאם והחזירו את הקבוצה החדשה מהפונקציה.

□ הפרש סימטרי (symmetricDifference):

יוצר קבוצה חדשה המכילה את כל מה ששייך לקבוצה A או לקבוצה B ולא שייך לחיתוך שלהן. כלומר המתודה תקבל שני פרמטרים A ו-B שהינם מופעים של SimpleSet שנרצה לבצע הפרש סימטרי ביניהם, היא תיצור קבוצה חדשה המכילה את כל האלמנטים בקבוצות A ו-B ללא האלמנטים השייכים לחיתוך בין הקבוצות, ניתן להשתמש במתודה add שכתבתם על מנת לעדכן את כל השדות בהתאם והחזירו את הקבוצה החדשה מהפונקציה.

□ קבוצת חזקה (powerSet):

יוצר קבוצה חדשה המכילה את כל תתי הקבוצות של הקבוצה A. כלומר המתודה תקבל פרמטר יחיד A שהינו מופע של SimpleSet שנרצה לחשב את קבוצת החזקה שלו, היא תיצור קבוצה חדשה המכילה את כל תתי הקבוצות של A, ניתן להשתמש במתודה add שכתבתם על מנת לעדכן את כל השדות בהתאם והחזירו את הקבוצה החדשה מהפונקציה.

שימו לב הפעולות מתבצעות על קבוצות מאותו סוג גנרי

כתבו main שמדגים את כל המתודות שמימשותן במחלקות SimpleSet ו-SetUtils.

בהצלחה