כתבו מחלקה Bucket המייצגת דלי.

במחלקה יש שדה capacity שמייצג את הקיבולת – מספר הליטרים המקסימלי שניתן להכניס לדלי.

כמו כן, במחלקה יש שדה current שמציין כמה ליטרים בפועל נמצאים כרגע בדלי.

שדות אלה הינם בעל הרשאת גישה private, שמשמעותה היא שאין לשדות אלה גישה מחוץ למחלקה.

במחלקה צריכות להיות המתודות הבאות:

בנאי שיקבל פרמטר אחד: הקיבולת של הדלי. הבנאי יבצע השמה של הפרמטר לשדה המתאים, ובשדה current ישים את הערך 0.

String toString()

המתודה מחזירה מחרוזת המייצגת את אובייקט הדלי, על פי הפורמט הבא:

The current state of the bucket: {current} of {capacity} liters.

boolean isEmpty()

המתודה מחזירה תשובה האם הדלי ריק כרגע.

boolean isFull()

המתודה מחזירה תשובה האם הדלי מלא כרגע.

float percent()

מחזירה את אחוז הליטרים בדלי.

boolean addLiter()

מוסיפה ליטר אחד אל הדלי, אם הוא אינו מלא. אם הדלי מלא, המתודה מחזירה false.

int addLiters(int amount)

מוסיפה לדלי מספר נקוב של ליטרים, המועבר אל המתודה כפרמטר amount. אם אין מקום בדלי לכל הליטרים, יתווספו לדלי מספר הליטרים המקסמלי המתאפשר. המתודה תחזיר את כמות הליטרים שעבורם אין מקום. לדוגמה, אם בדלי חסרים 3 ליטרים, ומנסים להוסיף אליו 5, המתודה תחזיר 2.

boolean spill()

מרוקנת את כל תכולת הדלי. אם הדלי כבר ריק, המתודה מחזירה false.

boolean spill(int amount)

מרוקנת מהדלי מספר נקוב של ליטרים, המועבר אל המתודה כפמטר amount. אם אין מספיק מים כדי לשפוך את מלוא הכמות, כלום לא יישפך והמתודה תחזיר false.

void fill()

ממלאת את כל תכולת הדלי עד סופו.

void fill(Bucket other)

מקבלת כפרמטר אובייקט אחר של Bucket, וממלאת את הדלי הנוכחי באמצעות כל המים שנמצאים בדלי האחר. דוגמה, אם בדלי הנוכחי חסרים 7 ליטר מים כדי להיות מלא, ובדלי האחר ישנם 11 ליטר מים, אנו "מעבירים" 7 ליטרים מהדלי האחר אל הדלי הנוכחי.

void setCapacity(int capacity)

מקבלת כפרמטר מספר שלם וקובעת את אותו המספר כקיבולת של הדלי.

לאחר הגדרת המחלקה Bucket, כתבו תוכנית ראשית ובה הפונקציות הבאות

הפונקציה

void magic()

פותרת את החידה הבאה: יש ברשותכם שני דליים: אחד בעל קיבולת של 7 ליטר, השני בעל קיבולת של 5 ליטר. כמו כן, ישנה בריכה שממנה ניתן למלא את הדליים. עליכם למלא את הדלי הגדול ב-6 ליטרים בדיוק. על הפונקציה ליצור שני אובייקטים של Bucket בשמות b1 ו-b2. ב-b1 תהיה קיבולת של 7 ליטר. ב-b2 תהיה קיבולת של 5 ליטר. עליכם לכתוב לוגיקה שבסיומה ב-b1 יהיו בדיוק 6 ליטרים **מבלי להשתמש במתודות מספר 5,6,8 ו-11.**

הפונקציה

Bucket reduceAmountInBuckets (Bucket[] buckets, float max)

הפונקציה מקבלת שני פרמטרים: מערך של אובייקטים מהמחלקה Bucket ומשתנה max מסוג float.

הפונקציה תיצור אובייקט של דלי חדש. עבור כל דלי שאחוז המים בו גבוה מ-max, היא תעביר מאותו דלי, אל הדלי החדש, כמות מסוימת של ליטרים כך שאחוז המים בו יהיה נמוך מ-max.

לדוגמה: max = 60, ובדלי מסוים יש קיבולת של 13 וכמות ליטרים נוכחית של 10. אחוז המים הקיים בדלי לפיכך הוא 76.9, ולכן יש להעביר ממנו מים אל דלי אחר עד שיוותרו בו פחות מ-60 אחוז.

הפונקציה

int[] fillExactAmount(Bucket[] buckets, int amount)

הפונקציה מקבלת שני פרמטרים: מערך של אובייקטים מהמחלקה Bucket ומספר amount מסוג int.

הפונקציה תבדוק האם ישנה קבוצה של דליים, שמלאים לפחות בחציים, שאם נשפוך את **כל** תכולת המים שבהם, נגיע בדיוק לערך של amount.

זאת אומרת, הערך של amount הוא 18, אז אנחנו צריכים לבדוק האם ניתן למלא בדיוק 18 ליטרים באמצעות שפיכת כל המים הנמצאים בדליים הרלוונטיים שבמערך. אם קיימת קבוצה כזו, הפונקציה תחזיר מערך של intים, שהוא האינדקסים של הדליים הרלוונטיים.

הפונקציה

void playBucketsGame()

הפונקציה תשתמש במחלקה Bucket לצורך משחק פשוט. במשחק זה, השחקן מקבל שלושה דליים, בכל אחד מהם קיבולת אקראית עד 20 וכמות ליטרים התחלתית אקראית. בנוסף, הוא מקבל דלי ריק נוסף עם קיבולת אקראית. בכל דלי חייבת להיות קיבולת אחרת. המטרה של השחקן היא למלא את אחד הדליים בכמות אקראית מסוימת. לדוגמה, שחקן שמקבל דלי עם 10/13 ליטרים, דלי עם 4/8 ליטרים, וצריך למלא את אחד הדליים ב-9 ליטרים.

התוכנית תבדוק האם הוא הצליח במשימתו, ותכתוב בתוך כמה מהלכים עשה זאת. **במשחק אין** **להשתמש במתודות מספר 5,6,8 ו-11.**