

חזרה בנושא אוספים ספציפיים

בנק - Bank

מטרות

כתיבת מחלקה המגדירה אוסף ייחודי (ספציפי), בחירת ייצוג מתאים. תרגול של עצמים מורכבים, איברי מחלקה, רשימה ועוד.

רמת השאלה

השאלה מסכמת נושאים רבים שנלמדו ביחידה. מתאימה לתרגיל מעבדה. ניתן להשתמש בחלקים לצורך מבחן (ראו בהנחיות המיוחדות).

חלק א

המחלקה Bank

מגדירה סניף בנק שמנהל אוסף חשבונות בנק. בכל חשבון בנק נשמרים הנתונים: מספר-חשבון (המזהה את החשבון באופן ייחודי), שם בעל החשבון ויתרת הכסף בחשבון. לכל החשבונות יש מסגרת אשראי קבועה אחת בגובה 5000 שקלים.

Bank()	הפעולה בונה סניף בנק חדש
int openAccount (String ownerName)	הפעולה פותחת חשבון בנק חדש שמספרו נקבע על ידי המחלקה בצורה אוטומטית-סידרתית החל מהמספר 1000. שם בעל החשבון הוא ownerName ויתרת הכסף בחשבון היא 0 שקלים. חשבון זה מתווסף לאוסף החשבונות בסניף הבנק. הפעולה מחזירה את מספר החשבון שנפתח
void deposit (int accountNum, double amount)	הפעולה מפקידה amount שקלים לחשבון הבנק שמספרו accountNum. הנחות: קיים חשבון שמספרו accountNum. amount גדול מאפס.
boolean withdrawal (int accountNum, double amount)	הפעולה מושכת amount שקלים מחשבון accountNum. הפעולה תחזיר true אם המשיכה לא גרמה לחרیגה מהמסגרת, אחרת תחזיר false והמשיכה לא תתבצע. הנחות: קיים חשבון שמספרו accountNum. amount גדול מאפס.
double getBalance (int accountNum)	הפעולה מחזירה את היתרה ב-accountNum. הנחה: קיים חשבון שמספרו accountNum.
String getOwnerName (int accountNum)	הפעולה מחזירה את שם בעל החשבון accountNum. הנחה: קיים חשבון שמספרו accountNum.
boolean accountExist (int accountNum)	הפעולה מחזירה true אם accountNum קיים בסניף הבנק, אחרת תחזיר false
List<Integer> getAllAccountsNums()	הפעולה מחזירה את אוסף כל מספרי החשבונות הקיימים בסניף זה.

<code>String toString()</code>	<p>הפעולה מחזירה מחרוזת המתארת את סניף הבנק :</p> <p><accNum1>, <ownerName1>, <balance1> <accNum2>, <ownerName2>, <balance2> :</p>
--------------------------------	--

מה עליכם לעשות?

1. כתבו את המחלקה Bank ותעדו אותה. ניתן להשתמש במחלקות עזר נוספות.
2. כתבו תוכנית בדיקה TestBank.java.
3. הוסיפו לתוכנית הבדיקה את הפעולה הבאה :

<code>static void printNegBalance (Bank bank)</code>	<p>הפעולה תדפיס את כל החשבונות בסניף bank שיתרתם שלילית. כל חשבון יודפס בשורה נפרדת בה יופיעו הנתונים : מספר החשבון, שם בעל החשבון ויתרת הכסף בחשבון</p>
--	--

4. כפי שניתן לשים לב המחלקה Bank מאפשרת פתיחת מספר חשבונות ששם בעליהם יכול להיות זהה. הוסיפו למחלקה Bank את הפעולה הבאה :

<code>int[] getAllAccountsOf (String name)</code>	<p>הפעולה תחזיר מערך חד-ממדי בו יופיעו כל מספרי החשבונות ששם בעליהם הוא name.</p>
---	---

חלק ב'

בצעו את השינויים הנדרשים במחלקה Bank שכתבתם בשלב א' כך שבעת זימון הפעולה `toString()` תוחזר מחרוזת המתארת את פרטי כל החשבונות בסניף הבנק (כפי שהיה קודם), ובנוסף, עבור כל חשבון תפורט כל היסטורית פעולות המשיכה וההפקדה (דו"ח תנועות), שבוצעו בו מאז שנפתח ועד לרגע זימון פעולה זו. סדר היסטוריית הפעולות יופיע על פי סדר ביצוען.

הנחיות מיוחדות

1. כאשר נבנה סניף בנק חדש הוא למעשה סניף שאין בו חשבונות (סניף "ריק").
2. ניתן לחסוך בשלב א ולתת לתלמידים תוכנית בדיקה מוכנה.
3. הגדרת התכונות מתרגלת שימוש בסוגי איברים שונים : מסגרת האשראי היא תכונת מחלקה ולכן היא סטטית. כמו כן עליה להיות מוגדרת כ-`final`. מספר החשבון גם הוא תכונת מחלקה סטטית.
4. האוסף המדובר הוא אוסף ספציפי. כלומר הוא אינו אוסף גנרי שיכול לשמש לאחסון של איברים שונים אלא רק חשבונות בנק.
5. יש להיעזר במחלקה Account המגדירה חשבון יחיד. במחלקה Bank יש להגדיר אוסף של חשבונות. מאחר ואין מגבלה על כמות החשבונות שניתן לפתוח, רצוי להשתמש ברשימה או מפה. השימוש במפה עדיף מכיוון שהגדרת המשימה היא שאין כפילות של חשבונות בעלי אותו מספר. שימו לב שאין חשיבות לסדר שמירת החשבונות באוסף (סיבה נוספת להתאמת השימוש במפה).
6. תרגיל זה בשלמותו מיועד לתרגול במעבדה למשך כשעתיים. במידה ורוצים לתת אותו כתרגיל במבחן יש להסתפק בסעיפים בודדים ואת השאר לתת לתלמידים בהתאמה.

הרחבות אפשריות

1. אפשר להוסיף למחלקה Bank פעולה להעברת כסף מחשבון לחשבון (moneyTransfer).
2. אפשר להוסיף למחלקה Bank פעולה לסגירת או מחיקת חשבון קיים.
3. אפשר להרחיב את הרעיון של חלק ב' וליצור מצב בו יש לשמור את היסטורית הפעולות במבנה נתונים כגון מחסנית שיצריך ביטול של פעולה אחרונה שהתבצעה כדיכול בטעות.
4. ניתן להגדיר מספר סניפי בנק ולבצע פעולות על סניפי הבנק: למצוא איזה סניף הכי רווחי? (על פי סכום הכספים בכל היתרות או/וגם על פי מספר החשבונות שנפתחו בסניף).
5. ניתן להוסיף אילוץ נוסף המגדיר סדר של אוסף החשבונות. לדוגמה, ניתן לדרוש שהפעולה toString() תחזיר את כל החשבונות מסודרים בסדר עולה על פי ערך היתרה בהם. במצב זה על התלמיד להחליט אם הוא שומר את האוסף ממוין בכל שלב ושלב או שהוא מבצע מיון רק כאשר toString() מזומנת. ניתן לקיים דיון ביעילות שתי האפשרויות.