

הנדסת תוכנה – תרגיל בית 1

דגשים להגשת המטלה

1. תאריך הגשה: יום ראשון 06.06.2021 בשעה 23:59.
2. הגשה בזוגות בלבד!
3. הקוד חייב להיכתב בהתאם למוסכמות כתיבת הקוד בקורס כלל תיעוד כנדרש. קוד שלא עומד בדרישות יגרור הורדת ניקוד.
4. ההגשה מתבצעת ב-Moodle באזור המיועד על ידי אחד מהשותפים, לאחר יצירת קבוצה.
5. כל איחור בהגשה יגרור הורדת 20% מהציון בכל יום.
6. פורמט הגשת התרגיל נמצא בקובץ ההנחיות ב-Moodle. כל חריגה מפורמט זה תגרור ציון 0.
7. שאלות והבהרות ייכתבו בפורום ייעודי שייפתח לתרגיל הבית ב-Moodle.

מטרת התרגיל

עבודה ראשונה עם מחלקות.

הכנות טרם תחילת התרגיל

1. פתיחת פרויקט Java חדש.
2. הורדת קבצי התרגיל, והעתקת הקובץ Main.java בלבד אל תוך תיקיית ה-src.

הוראות כלליות

1. יש לבדוק שהקוד עובר הידור (קומפילציה) ללא שגיאות.
2. מומלץ להריץ את התוכנית עם מספר קלטים שונים ולחשוב על מקרי קצה אפשריים.
3. מומלץ לחזור על התרגולים וההרצאות וכן להיעזר באינטרנט.
4. מומלץ להשתמש ב-Git במהלך כתיבת התרגיל.
5. יש לתעד את המחלקות ואת הפעולות שבהן, לפי מדריך התיעוד שבאתר הקורס.

הוראות הגשה

1. יש למלא את הוראות ההגשה בהתאם למסמך הדרישות "הנחיות כלליות לפתרון והגשת תרגילי הבית" אשר מופיע באתר הקורס.
2. הגשה אלקטרונית בלבד דרך אתר הקורס ב-moodle. ההגשה תכלול את קובץ ה-zip בלבד.
3. ההגשה מתבצעת על ידי אחד מבני הזוג לאחר שיצר קבוצה ובן הזוג השני הצטרף אליה.
4. תרגיל בית שלא יוגש על פי הוראות ההגשה – לא ייבדק.
5. יש להקפיד על יושרת הכנת התרגיל וההגשה.
6. יש לוודא כי הקוד מתקמפל – קוד אשר לא יעבור הידור יקבל ציון 0.
7. אין צורך להגיש את קבצי הקלט והפלט אשר ניתנים כזלק מתרגיל זה.

בתרגיל זה נבנה את משחק הקלפים "מלחמה".

במשחק זה משתתפים שני שחקנים, והשחקן המנצח הוא האחרון אשר נשארו בידיו קלפים.

הכנות למשחק

תחילה, מערבבים את חפיסת הקלפים באופן רנדומלי, והקלפים מחולקים שווה בשווה בין שני המשתתפים על ידי מתן קלף אחד לכל משתתף (לפי סדר אלפביתי של שמותיהם) עד אשר לא נשארים קלפים בחפיסה.

מהלך המשחק

המשחק מורכב ממספר סבבים. בכל סבב שני המשתתף שולפים את הקלף העליון מחפיסתם (לפי סדר אלפביתי של שמותיהם), ושמים אותו בערימה מרכזית. המשתתף אשר שלף את הקלף בעל המספר הגבוה לוקח את הקלפים, ושם אותם בערימה נפרדת משלו (השונה מערימת הקלפים איתה הוא משחק). במידה ושני השחקנים הניחו את אותו הקלף, הם יבצעו ביניהם "מלחמה": שליפה של שני קלפים ולאחריהם קלף שלישי. הזוכה במלחמה הוא זה ששלף את הקלף השלישי הגבוה ביותר, והוא לוקח אליו את כל הקלפים של הסבב. במידה וגם במצב זה שני המשתתפים שלפו את אותו הקלף, הם יבצעו ביניהם מלחמה נוספת (וחוזר חלילה).

בכל פעם שמשתתף צריך לשלוף קלף אך חפיסת הקלפים אשר בידו נגמרה, עליו לערבב את החפיסה השנייה שלו (חפיסת הקלפים בהם זכה), ולהפוך את ערימה זו לערימת המשחק שלו. כאשר יזכה בקלפים נוספים, הוא ישים אותם שוב בערימה נפרדת.

במידה ולמשתתף נגמרים הקלפים גם בחפיסה השנייה שלו, הוא יוצא מן המשחק, ומפסיד בו.

מימוש המשחק

על מנת לממש את המשחק, תצטרכו ליצור מספר מחלקות. מחלקות אלו יתבססו אחת על השנייה, וביחד יהוו את מכלול המשחק.

ראשית, עליכם ליצור enum בשם Shape אשר ייצג צורה של קלף. לכל קלף קיימות ארבע צורות אפשריות:

- עלה (Spades), אשר מסומן בתו ♠.
- יהלום (Diamonds), אשר מסומן בתו ♦.
- תלתן (Clubs), אשר מסומן בתו ♣.
- לב (Hearts), אשר מסומן בתו ♥.

הערה: enum הוא טיפוס נתונים אשר מאפשר לנו לשמור קבועים בעלי שם בצורה נוחה, מבלי צורך להתעסק עם ערכים מפורשים. ניתן לקרוא עוד על enum [בקישור זה](#).

לאחר מכן, עליכם להגדיר מספר מחלקות נוספות.

מחלקת Card

מחלקה זו מייצגת קלף בודד. לכל קלף משויך מספר שלם בין 1 ל-13, כאשר 1 מייצג אס, 11 מייצג נסיך, 12 מייצג מלכה, ו-13 מייצג מלך. לקלפים בעלי אחד מן הערכים הנ"ל נאמר כי הם מייצגים תמונה. בנוסף, לכל קלף משויכת צורה כלשהי מבין ארבעת הצורות המוגדרות ב-enum. ידוע כי לאחר שנקבעו לקלף מספרו וצורתו, לא ניתן לשנותם.

על המחלקה לכלול:

- תכונות מתאימות.
- פעולות get ו-set לתכונות הנדרשות בכך.
- בנאי המקבל את כל התכונות ומאתחל אותן.
- פעולה `int compare(Card other)` אשר מחזירה מינוס 1 אם המספר של הקלף הנוכחי קטן מן המספר של `other`, 0 אם המספרים שווים, ו-1 אם המספר של הקלף הנוכחי גדול יותר.
- פעולה `String toString()` אשר מחזירה מחרוזת המייצגת את הקלף, בפורמט הבא:
`val of shape`, כאשר `val` הוא ערך הקלף, ו-`shape` הוא צורתו של הקלף. יש לייצג את צורת הקלף באמצעות התו המתאים לה. בנוסף, במידה והקלף מייצג תמונה, יש לרשום את התמונה אותה הוא מייצג במקום את המספר. לדוגמה, הקלף שערס 12 וצורתו לב מיוצג על ידי המחרוזת "Queen of ♥", והקלף שערס 5 בצורתו תלתן מיוצג על ידי המחרוזת "5 of ♣".

מחלקת Deck

מחלקה זו מייצגת חפיסת קלפים. חפיסת קלפים מכילה רשימה של קלפים.

על המחלקה לכלול:

- תכונות מתאימות.
- בנאי אשר מקבל פרמטר בוליאני. במידה וערכו `true`, יש לאתחל את רשימת הקלפים וליצור חבילת קלפים מלאה (סך הכל 52 קלפים, ללא ג'וקר). את החבילה יש ליצור באופן הבא: ראשית, ניצור את הקלפים שצורתם עולה, בסדר עולה (מ-1 עד 13). לאחר מכן, יש ליצור בסדר עולה את הקלפים שצורתם יהלום, ואז הקלפים שצורתם תלתן, ולבסוף את הקלפים שצורתם לב. במידה וערך הפרמטר `false`, יש להשאיר את חפיסת הקלפים ריקה.
- פעולה `void addCard(Card card)` אשר מקבלת קלף, ומוסיפה אותו בתור הקלף העליון בחפיסה.
- פעולה `Card removeTopCard()` אשר מסירה את הקלף האחרון מן החפיסה ומחזירה אותו.
- פעולה `boolean isEmpty()` אשר מחזירה האם חפיסת הקלפים ריקה.
- פעולה `void shuffle()` אשר מערבבת את חפיסת הקלפים. ערבוב החפיסה יתבצע על באופן הבא: ניצור לולאה אשר תרוץ 50 פעמים. בכל איטרציה נגריל שני אינדקסים מן המערך, ונחליף בין הערכים אשר נמצאים באינדקסים הללו. לצורך הגרלת המספרים יש להשתמש במשתנה הסטטי `rnd` אשר נמצא במחלקה `Main`. ניתן לקרוא [בקישור זה](#) על הגרלת מספרים שלמים בטווח מסוים.

מחלקת Player

מחלקה זו מייצגת שחקן במשחק. שחקן מאופיין על ידי שמו. בנוסף, לשחקן יש שתי חפיסות קלפים: חפיסת משחק ממנה הוא שולף קלפים, וחפיסת זכייה, בה הוא שומר את הקלפים שזכה בהם (עד אשר חפיסת המשחק שלו מתרוקנת).

על המחלקה לכלול:

- תכונות מתאימות.
- פעולות `get`-ו-`set` לתכונות הנדרשות בכך.
- בנאי אשר מקבל את שם השחקן, ומאתחל את שתי חפיסות הקלפים שלו לחפיסות ריקות.
- פעולות אשר מוסיפות קלף לכל אחת מחפיסות הקלפים.
- פעולה `drawCard()` לשליפת קלף באופן המתואר בסיפור.
- פעולה `boolean outOfCards()` אשר בודקת האם לשחקן נגמרו כל הקלפים.
- פעולה `String toString()` אשר מחזירה את שמו של השחקן.

מחלקת WarGame

מחלקה זו מייצגת משחק מלחמה. כל משחק מורכב משני שחקנים, ומחפיסת קלפים עבור כל שחקן. חפיסות הקלפים נועדו לצורך אחסון זמני של הקלפים אשר המשתתפים שלפו מחפיסתם.

על המחלקה לכלול:

- תכונות מתאימות.
- פעולות `get`-ו-`set` לתכונות הנדרשות בכך.
- בנאי אשר מקבל את שמות שני השחקנים, ומאתחל את תכונות המחלקה. ניתן להניח כי לשני השחקנים יש שמות שונים.
- פעולה `void initializeGame()` אשר מחלקת את הקלפים לשחקנים באופן המתואר בסיפור.
- פעולה `String start()` אשר מאתחלת את המשחק, משחקת אותו ומחזירה את המנצח.

הדפסות במהלך המשחק

- בעת אתחול המשחק, יש להדפיס את ההודעה `Initializing the game...`
- בכל פעם אשר משתתף שולף קלף, פרט למקרה בו קלף זה הוא אחד מבין שני הקלפים הראשונים במלחמה, יש להדפיס את שמו ואת הקלף אשר הוא הוציא. לדוגמה, במידה ושחקן בשם דני הוציא קלף 5 יהלום, תודפס ההודעה `"Dani drew 5 of ♦"`.
- במידה ויש מלחמה, יש להדפיס בתחילתה את ההודעה `Starting a war...` בנוסף, במידה והמלחמה לא נקטעת באמצע עקב זה שלאחד השחקנים נגמרו כל הקלפים, יש להדפיס בסופה את ההודעה `name won the war`, כאשר `name` הוא שמו של השחקן שזכה במלחמה.
- כאשר המשתתפים שולפים את שני קלפי המלחמה הראשונים, יש להדפיס את ההודעה `name drew a war card`, כאשר `name` זהו שמו של המשתתף אשר שלף את הקלף.
- במידה ואין כלל מלחמה בסבב כלשהו, יש להדפיס את שם השחקן שניצח בסבב (השחקן שלקח אליו את הקלפים) `name won`, כאשר `name` זהו שמו של אותו שחקן.
- בתחילת כל סבב חדש במשחק יש להדפיס "קו הפרדה" בפורמט הבא:
----- Round number n ----- כאשר `n` הוא מספר הסבב.

הערות

- יש ליצור כל מחלקה (ואת ה-enum) בקובץ נפרד.
- בכל מחלקה ניתן להוסיף פעולות ותכונות נוספות.
- במהלך כתיבת הקוד, יש לשמור על עקרונות תכנות נכונים, כפי שנלמד בקורס.
- במצב של מלחמה, השחקנים ישלפו את הקלפים הדרושים למלחמה בזה אחר זה, קלף אחד בכל פעם. לדוגמה, במקרה שבו יש מלחמה בין דני ויוסי" תחילה דני ישלף קלף אחד, ואז יוסי ישלף קלף אחד. לאחר מכן דני ישלף את הקלף השני ויוסי ישלף את הקלף השני, ולבסוף דני ישלף את הקלף השלישי ויוסי ישלף את הקלף השלישי.
- יש להקפיד על סדר יצירת חפיסת הקלפים וחלוקתה בין המשתתפים ועל סדר המשתתפים במשחק. אי הקפדה על הסדר יוביל לשגיאה בביצוע ההגרלות, דבר אשר יפגע במהלך המשחק ובפלט התקין.

הרצת התוכנית וביצוע בדיקות

בפעולה הראשית קיים קטע קוד אשר משמש לביצוע בדיקת לקוד שכתבתם. הבדיקות משתמשות בקובץ HW1_input.txt אשר מסופק לכם ביחד עם תרגיל זה.

על מנת להריץ את התוכנית, יש לספק את הנתיב לקובץ זה כפרמטר.

להזכרכם, חלק מן הבדיקה נעשה באופן אוטומטי, ולכן אין לשנות את הפעולה הראשית, ובפרט אין לשנות את פעולת ההדפסה המתבצעות בה.

מצורף לתרגיל זה קובץ הפלט HW1_output.txt, על מנת שתוכלו לבצע את ההשוואה באופן ידני (או על ידי שימוש ב-DiffMerge).

אתר הבדיקה יהיה מוכן בקרוב לשימושכם.

שימו לב: העלאת הקובץ לאתר הבדיקה אינה מחליפה את ההגשה באתר הקורס. חובה להגיש את העבודה בתיבה הייעודית לכך ב-moodle.

בהצלחה.