



## הנדסת תוכנה – תרגיל בית 0

### דגשים להגשת המטלה

1. **תאריך הגשה: יום שלישי 02.05.2023 בשעה 23:59.**
2. **הגשה בזוגות בלבד!**
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. פורמט הגשת התרגיל נמצא בקובץ ההנחיות ב-Moodle. **כל חריגה מפורמט זה תגרור ציון 0.**

### מטרת התרגיל

היכרות עם סביבת הפיתוח IntelliJ IDEA והתנסות ראשונית בג'אווה.

### הכנות טרם תחילת התרגיל

1. התקנת סביבת העבודה IntelliJ IDEA בהתאם לקובץ ההנחיות "intellij – הוראות" אשר מופיע באתר הקורס.
2. פתיחת פרויקט ג'אווה חדש. הנחיות לפתיחת פרויקט והכנת סביבת העבודה ניתן למצוא באתר של JetBrains.
3. 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 קליטים שונים ולחשוב על מקרי קצה אפשריים.
3. מומלץ לחזור על התרגולים וההרצאות וכן להיעזר באינטרנט.
4. יש להשתמש בגרסה 9.0.4 של ג'אווה בעת פתרון התרגיל.
5. מומלץ להשתמש ב-Git במהלך כתיבת התרגיל.

### 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 קובץ ה-zip בפורמט `HW0_<id1>_<id2>` כאשר `<id1>` ו-`<id2>` הם תעודות הזהות של המגשים.
3. ההגשה מתבצעת על ידי אחד מבני הזוג לאחר שיצר קבוצה וכן 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 אפס.
7. אין צורך להגיש את קבצי הקלט והפלט אשר ניתנים כחלק מתרגיל זה.



## משחק הצוללות

בתרגיל בית זה תממשו משחק לשני שחקנים הנקרא "צוללות".

במשחק הצוללות לכל שחקן יש שני לוחות בגודל  $n \times m$  – לוח משחק ולוח ניחושים. בנוסף, לכל שחקן ישנו מספר כלשהו של צוללות בגדלים שונים, אותן הוא ממקם על גבי לוח המשחק שלו, על פי החוקים שיתוארו בהמשך. לאחר ששני השחקנים מיקמו את כל הצוללות שלהם, המשחק מתחיל, וכל שחקן מנסה להטביע את הצוללות של היריב. בכל תור, שחקן אחד מנחש משבצת אותה הוא עוד לא ניחש ואומר אותה ליריבו. לאחר מכן, השחקן השני אומר לו האם הוא פגע בצוללת שלו (במשבצת אכן יש צוללת) או שהוא פספס (אין צוללת במשבצת). במידה והצוללת נפגעה כולה היריב מוסיף ואומר שהצוללת הוטבעה. בהתאם לתגובתו של היריב (פגיעה או פספוס), השחקן מסמן את המשבצת על גבי לוח הניחושים שלו. לאחר מכן התור מסתיים ומתחיל התור של השחקן היריב, אשר פועל לפי אותם החוקים. המשחק מסתיים כאשר כל הצוללות של אחד מהשחקנים הוטבעו, והמנצח הוא השחקן שנשארו לו צוללות שלא הוטבעו.

בתרגיל זה תממשו את משחק הצוללות, עבור משתמש המתחרה מול מחשב.

## מימוש המשחק

כעת נעבור למימוש המשחק. במהלך המשחק יהיה עליכם לקלוט מידע מהמשתמש. לצורך כך הוגדר בקובץ Main.java משתנה מטיפוס Scanner בשם scanner אשר ישמש אתכם לאורך התרגיל.

### גודל הלוח, מספר הצוללות וגודליהן

לפני תחילת המשחק, יהיה עליכם לקלוט מן המשתמש את גודל הלוחות המשחק. עליכם להדפיס את ההודעה "Enter the board size", ולאחריה לקלוט מן המשתמש את גודל הלוחות. הקלט יוכנס בפורמט מחרוזת "nXm", כאשר n מייצג את מספר השורות בכל לוח ו-m מייצג את מספר העמודות.

לאחר מכן תעברו לקלוט את גודלי הצוללות וכמות הצוללות מכל גודל. על מנת לבצע זאת, עליכם להדפיס את ההודעה "Enter the battleships sizes". לאחר מכן עליכם לקלוט את המידע מן המשתמש, אשר יוכנס בפורמט "n1Xs1 n2Xs2 ... nkXsk" כאשר k מציין את מספר הגדלים השונים של הצוללות, si מציין את הגודל ה-i ו-ni מציין את מספר הצוללות בגודל si.

לדוגמה, במידה ויוזן הקלט "1X2 2X3 2X4", לכל שחקן יהיו 5 צוללות: צוללת אחת בגודל 2, 2 צוללות בגודל 3, ו-2 צוללות בגודל 4.

## מיקום צוללות המשתמש

לאחר שהמשתמש בחר את הכמות והגודל של הצוללות, עליו למקם אותן על גבי לוח המשחק שלו. מיקום הצוללות יתבצע לפי סדר עולה של גודלן. בעבור כל צוללת, יש להדפיס את ההודעה "Enter location and orientation for battleship of size s" כאשר s מציין את גודלה של הצוללת. לאחר מכן עליכם לקלוט מן המשתמש את המיקום של הצוללת ואת האוריינטציה שלה. הקלט יוכנס בפורמט "x, y, orientation" כאשר orientation מייצג את כיוונה של הצוללת, ויכול להיות 0 עבור צוללת אופקית או 1 עבור צוללת אנכית, ו-x, y מייצגים את המשבצת ממנה מיקום הצוללת יתחיל. צוללת אופקית בגודל s תמוקם במשבצות (x, y) עד (x, y+s-1), וצוללת אנכית בגודל s תמוקם במשבצות (x, y) עד (x+s-1, y).

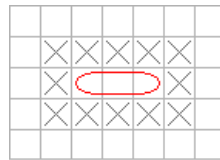


## הטכניון – מכון טכנולוגי לישראל הפקולטה למדעי הנדסה והחלשות הנדסת תוכנה אביב תשפ"ג



בעת מיקום הצוללות, עליכם לאכוף תקינות:

- על הכיוון של הצוללות להיות 0 או 1 בלבד. במידה ויתקבל ערך אחר, תודפס ההודעה "Illegal orientation, try again!"
  - על המשבצת שהתקבלה להיות בתוך גבולות הלוח. במידה והיא מחוץ ללוח תודפס ההודעה "Illegal tile, try again!"
  - צוללת אינה יכולה לחרוג מגבולות הלוח. במידה והיא חורגת ממנו תודפס ההודעה "Battleship exceeds the boundaries of the board, try again!" שימו לב שהנקודה הקודמת עסקה רק במשבצת ההתחלתית של הצוללת, ואילו נקודה זו מתייחסת לכל המשבצות בהן הצוללת תמוקם.
  - צוללת אינה יכולה להיות חופפת עם צוללת אחרת שכבר ממוקמת על גבי הלוח. במידה וזה קורה תודפס ההודעה "Battleship overlaps another battleship, try again!"
  - צוללת אינה יכולה להיות סמוכה לצוללת אחרת שכבר ממוקמת על גבי הלוח, כולל באלכסון – יש לשמור על מרווח של לפחות משבצת אחת ריקה בין כל שתי צוללות. במידה וישנה צוללת סמוכה, תודפס ההודעה "Adjacent battleship detected, try again!"
- לצורך המחשה, בתמונה מוצגת צוללת ומסביבה מסומנת משבצות בהן לא יכולה להיות צוללות נוספות בעקבות הצורך לשמור על מרווח.

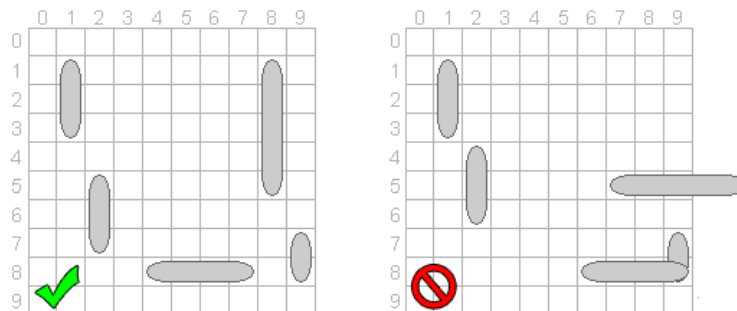


יש לבדוק את תקינות הקלט לפי הסדר המצוין ברשימה. במקרה של קלט לא תקין, יש להדפיס לכל היותר הודעת שגיאה אחת ומיד לאחר מכן לקלוט מחדש את המיקום והכיוון של הצוללת (מבלי להדפיס את ההודעה הראשונה), לבדוק את תקינותו ולחזור על התהליך עד לקבלת קלט תקין.

לפני הקלט הראשון ולאחר כל קלט תקין יש להדפיס את ההודעה "Your current game board:" ולאחריה את לוח המשחק של המשתמש, כאשר כל משבצת בה אין צוללת מיוצגת על ידי קו מפריד ('-') (שימו לב שזה לא מקף) וכל משבצת בה יש צוללת מיוצגת על ידי סולמית ('#'). פרטים נוספים על אופן הדפסת הלוח מופיעים בהמשך.

### מיקום צוללות המחשב

מיקומי הצוללות של המחשב יוגרלו באופן דומה לאופן בו המשתמש בוחר את מיקומי הצוללות שלו. ההגרלות יתבצעו בסדר עולה של גודל הצוללות. בעבור כל צוללת, יתבצעו שלוש הגרלות: הגרלה ראשונה של מספר שלם בטווח  $[0, n]$  עבור  $x$ , הגרלה של מספר שלם בטווח  $[0, m]$  עבור  $y$  והגרלה של מספר שלם בטווח  $[0, 2]$  עבור הכיוון של הצוללת, כאשר 0 מציינת צוללת אופקית ו-1 מציינת צוללת אנכית. לאחר ביצוע שלוש ההגרלות יש לבדוק את תקינות הצוללת באותו האופן שבו בודקים את תקינות הצוללות של המשתמש, אך מבלי להדפיס הודעת שגיאה. במידה והצוללת אינה תקינה, יש לבצע את כל שלוש ההגרלות פעם נוספת, לבדוק את תקינותו וחזור חלילה. לצורך ביצוע ההגרלות, עליכם להשתמש במשתנה `rnd` המוגדר בקובץ `Main.java`.



בתמונה השמאלית מופע מיקום חוקי של צוללות על גבי לוח משחק בגודל  $10 \times 10$ , ובתמונה הימנית מופיע מיקום שאינו חוקי.



## הטכניון – מכון טכנולוגי לישראל הפקולטה למדעי הנדסה ותוכנה הנדסת תוכנה אביב תשפ"ג



### ביצוע התקפות – תור המשתמש

התור הראשון במשחק הוא של המשתמש.

בכל תור של המשתמש, (כולל בתור הראשון) יש להדפיס תחילה את ההודעה *"Your current guessing board: "* ולאחריה את לוח הניחושים שלו, כאשר כל משבצת שהמשתמש עדיין לא ניחש אותה מיוצגת על ידי קו מפריד ('-'), כל משבצת בה יש פגיעה מיוצגת על ידי וי ('V') וכל משבצת בה יש פספוס מיוצגת על ידי איקס ('X').

לאחר מכן יש להדפיס את ההודעה *"Enter a tile to attack"* ולקלוט מן המשתמש את המשבצת אותה הוא רוצה לתקוף. הקלט יוכנס בפורמט *"x, y"*. במקרה והקלט חורג מגבולות הלוח יש להדפיס *"Illegal tile, try again!"*, ובמידה והמשבצת שנקלטה כבר הותקפה יש להדפיס *"Tile already attacked, try again!"*. במקרה של קלט לא תקין יש לחזור על תהליך הקליטה עד לקבלת משבצת תקינה.

במידה ותקיפה זו הייתה פספוס, יש להדפיס *"That is a miss!"*. במידה והתקיפה הצליחה יש להדפיס *"That is a hit!"* ובמידה והתקיפה גרמה לטביעת הצוללת יש להדפיס בנוסף *"The computer's battleship has been drowned, r more battleships to go!"*, כאשר  $r$  מייצג את מספר הצוללות שנשארו למחשב.

### ביצוע התקפות – תור המחשב

בתורו של המחשב יש להגריל את מיקום התקיפה על ידי ביצוע שתי הגרלות, באופן דומה להגרלות שבוצעו על מנת למקם את הצוללות של המחשב (ללא ההגרלה השלישית של כיוון הצוללות). במידה והמשבצת שהוגרלה כבר הותקפה על ידי המחשב יש לחזור על שתי ההגרלות מבלי להדפיס הודעת שגיאה עד לקבלת משבצת תקינה.

לאחר מכן יש להדפיס *"The computer attacked (x, y)"* כאשר  $x, y$  זו המשבצת שהותקפה. בנוסף יש להדפיס את תוצאת ההתקפה באופן דומה להדפסה שמתבצעת לאחר תורו של המשתמש: במידה והתקיפה הייתה פספוס, יש להדפיס *"That is a miss!"*. במידה והתקיפה הצליחה יש להדפיס *"That is a hit!"*, ובמידה והתקיפה גרמה לטביעת הצוללת יש להדפיס בנוסף *"Your battleship has been drowned, you have left r more battleships!"*, כאשר  $r$  מייצג את מספר הצוללות שנשארו למשתמש.

לבסוף, יש להדפיס *"Your current game board: "* ולאחר מכן את לוח המשחק המעודכן של המשתמש, כאשר כל משבצת בה אין צוללת מיוצגת על ידי קו מפריד ('-'), כל משבצת בה יש צוללת ועדיין לא הותקפה מיוצגת על ידי סולמית ('#') וכל משבצת בה יש צוללת והיא הותקפה מיוצגת על ידי איקס ('X').

### סיום המשחק

כפי שצוין, המשחק מסתיים כאשר כל הצוללות של אחד מהשחקנים הוטבעו, והמנצח הוא השחקן שנשארו לו צוללות שלא הוטבעו.

בסיום המשחק, במידה והמשתמש ניצח יש להדפיס *"You won the game!"* ובמידה והמחשב ניצח יש להדפיס *"You lost:"*.



## הטכניון – מכון טכנולוגי לישראל הפקולטה למדעי הנדסה והחלטה הנדסת תוכנה אביב תשפ"ג



### הדפסת לוח

במהלך המשחק יש להדפיס את הלוחות השונים באופן תכוף. הדפסת לוח כלשהו (לוח משחק ולוח ניחוש) תתבצע באופן הבא:

ראשית, יש להדפיס שורה של מספרי העמודות, החל מ-0 ועד למספר העמודה האחרונה ( $m-1$ ) כאשר המספרים מופרדים על ידי רווח. שימו לב שיש להדפיס רווחים לפני הדפסת הספרה 0 על מנת ליישר את המספרים עם העמודות.

לאחר מכן יש להדפיס את שורות הלוח: לפני הדפסת השורה יש להדפיס את מספרה (מספר בין 0 ל- $n-1$ ), לאחריה להדפיס רווח ואז להדפיס את המשבצות של אותה השורה (בהתאם לתווים המתאימים שצוינו לאורך המסמך) מופרדות ברווח. במקרה הצורך, יש להדפיס רווחים לפני מספר השורה על מנת לשמור על כך שכל מספרי השורות יהיו מיושרים.

לבסוף, יש להדפיס שורה ריקה.

לדוגמה, כך נראית הדפסה של לוח משחק בגודל  $10 \times 11$  עליו מוקמו שלוש צוללות – אחת נותרה בשלמותה, אחת נפגעה חלקית ואחת הוטבעה:

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	-	-	-	-	-
1	-	#	X	#	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	#	-	-	-	-
4	-	-	-	-	-	#	-	-	-	-
5	-	-	-	-	-	#	-	-	-	-
6	-	-	-	-	-	#	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	X	X	X	X	X	-
10	-	-	-	-	-	-	-	-	-	-

שימו לב לרווחים המופיעים לפני מספרי השורות. כמות הרווחים נקבעת לפי השורה הנוכחית ולפי כמות הספרות בשורה האחרונה. לדוגמה, במידה ובלוח היו 101 שורות, לפני המספרים 0-9 יופיעו שני רווחים, לפני המספרים 10-99 יופיע רווח בודד ולפני המספר 100 לא יופיע רווח כלל. באופן דומה נקבע מספר הרווחים בשורה הראשונה, בה מופיעים רווחים לפני מספרי העמודות.

### הנחיות לפתרון

- בקובץ Main.java מוגדרת פעולה בשם battleshipGame. פעולה זו מהווה את נקודת ההתחלה של משחק צוללות בודד, והפעולה הראשית משתמשת בפעולה זו.
- בעת פתרון התרגיל יש להגדיר פעולות עזר נוספות. אין לתכנת את כל המשחק רק בפעולה battleshipGame.
- תרגיל זה עוסק בתכנות פרוצדורלי. יש לפתור אותו ללא יצירת מחלקות וללא הגדרת משתנים גלובליים.
- ניתן להניח כי הצוללות יוזנו לפי סדר עולה של גודלן.
- ניתן להניח כי קיים לפחות סידור חוקי אחד של הצוללות על גבי לוח המשחק.
- יש לתעד את כל הפעולות אותן אתם מגדירים על פי קובץ מוסכמות התיעוד אשר מופיע באתר הקורס.
- בעת פתרון התרגיל ניתן ואף מומלץ להגדיר קבועים ולא להשתמש במספרי קסם.
- יש להשתמש במשתנה rnd וב-scanner כפי שהם מוגדרים ואין לשנות את תוכנם או להגדירם מחדש.
- אין לייבא ספריות ואין להיעזר באף פעולה שלא אתם כתבתם, גם לא בפעולות אשר אינן דורשות ייבוא, פרט ל:

- פעולות הדפסה.
- פעולות של המשתנים rnd ו-scanner.
- פעולות על מחרוזות ופעולות המוגדרות במחלקה Integer.



#### הרצת התוכנית וביצוע בדיקות

בפעולה הראשית קיימים מספר קטעי קוד אשר משמשים לבדיקת הקוד. הבדיקות משתמשות בקובץ `HW0_input.txt` אשר מסופק לכם ביחד עם תרגיל זה. במהלך הבדיקות משוחקים 6 משחקים שונים.

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

להזכירכם, חלק מן הבדיקה נעשה באופן אוטומטי, ולכן אין לשנות את הפעולה הראשית ובפרט אין לשנות את פעולות ההדפסה המתבצעות.

על מנת לוודא כי הפעולות שכתבתם עובדות כראוי, מצורף לתרגיל זה קובץ הפלט `HW0_output.txt`, על מנת שתוכלו לבצע את ההשוואה באופן ידני (או על ידי שימוש ב-DiffMerge).

בהצלחה