

מטלה 1 – עקרונות תכנות מונחה עצמים

הנחיות הגשה: יש להגיש קובץ zip עם כל המחלקות הדרושות בתוכו בפורמט: "ID_ID" עם תעודות הזהות של שני הסטודנטים או "ID" במקרה של מגיש בודד.

חלק ראשון – 60 נק':

במטלה זאת עליכם לבנות את המשחק "שחמט ויקינגים". הסבר מדויק על המשחק נמצא בעמודים האחרונים. בכל מקרה נמליץ לכם תחילה לצפות בסרטון הבא (שימו לב שבסרטון לא מופיעים מקרי הקצה):

<https://fb.watch/mKM3NSnyPn/?mibextid=6AJuK9>

מצורפים לכם 6 קבצים,

שתי המחלקות: Main.java, GUI_for_chess_like_games.java,

שלושת הממשקים: PlayableLogic.java, Player.java, Piece.java,

וקובץ jar של המשחק לדוגמה, כפי שמצופה ממנו לעבוד.

שימו לב, את שאר המחלקות עליכם לכתוב בעצמכם לגמרי, ואילו בכל המחלקות הנתונות אסור לכם לשנות דבר (הקבצים לקריאה בלבד).

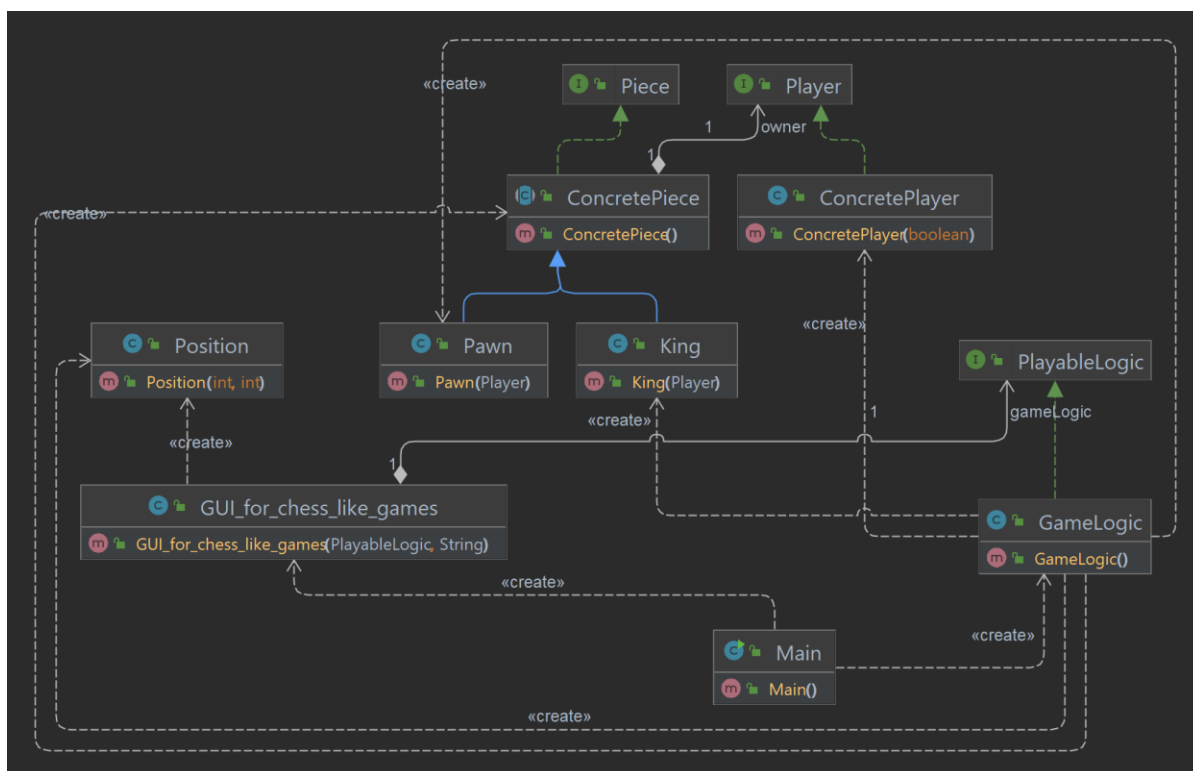
עליכם לכתוב ולממש את המחלקות הבאות תוך יישום עקרונות OOP שלמדנו:

• ConcretePiece - implements Piece	מחלקה אבסטרקטית המייצגת כלי בלוח
• Pawn - extends ConcretePiece	מחלקה המייצגת רגלי
• King - extends ConcretePiece	מחלקה המייצגת מלך
• Position	מחלקה המייצגת משבצת
• ConcretePlayer - implements Player	מחלקה המייצגת שחקן – מתקיף או מגן
• GameLogic - implements PlayableLogic	מחלקה עבור לוגיקת המשחק

הסיבה לשימוש במחלקה האבסטרקטית, היא שרוב מוחלט של המתודות, משותפות לשני סוגי הכלים. לכן נעדיף לממש אותן פעם אחת בלבד. במחלקות Pawn, King נממש רק את המתודות הייחודיות להן. חישבו איזה מתודות לממש איפה (למשל, מכיוון שמלך לא יכול לאכול אין סיבה להחזיק משתנה שסופר את הכלים שנאכלו במחלקה האבסטרקטית ולא רק בPawn).

להלן הסבר על הפונקציות המיישמות במחלקה GameLogic ותרשים UML של המחלקות, הממשקים והקשרים ביניהם:

שם המתודה	הסבר
<code>move(Position a, Position b)</code>	מנסה לבצע מהלך של כלי ממיקום אחד למיקום אחר על לוח המשחק. מחזיר <code>true</code> אם המהלך חוקי והצלחה, אחרת מחזיר <code>false</code> .
<code>getPieceAtPosition(Position position)</code>	מקבל את הכלי הנמצא במיקום שניתן על לוח המשחק. מחזיר <code>null</code> אם אין שם כלי.
<code>getFirstPlayer()</code>	מקבל את השחקן הראשון.
<code>getSecondPlayer()</code>	מקבל את השחקן השני.
<code>isGameFinished()</code>	בודק האם המשחק הסתיים, מחזיר <code>true</code> אם המשחק הסתיים, אחרת מחזיר <code>false</code> .
<code>isSecondPlayerTurn()</code>	בודק האם זהו תור השחקן השני כרגע.
<code>reset()</code>	מאתחל את המשחק למצבו ההתחלתי, כולל מחיקת הלוח והמידע על השחקנים.
<code>undoLastMove()</code>	מבטל את המהלך האחרון שבוצע במשחק, משחזר את מצב הלוח וסדר התורות ודואג לעדכן את כל הנתונים אצל כלל האובייקטים. (בונוס)
<code>getBoardSize()</code>	מקבל את גודל הלוח במשחק.



בנוס – חלק ראשון:

ממשו את כפתור ה"back". ממשו זאת כך שניתן יהיה לחזור תור אחרי תור, עד לתחילת המשחק הנוכחי.

חשבו על מבנה נתונים מתאים, ושימו לב לכך שחלק 2 אמור לעבוד כראוי לאחר השימוש בפיצ'ר הזה.

לאחר כל ניצחון של אחד מהצדדים, נרצה לקבל מידע סטטיסטי על המשחק שהסתיים. הדפסו את המידע על פי הסעיפים הבאים. אחרי כל סעיף הדפסו שורה של 75 כוכביות. דוגמא של פלט ממשחק בהמשך.

1. לאחר ניצחון של אחד הצדדים, הדפסו עבור כל כלי על הלוח, את היסטוריית הצעדים שלו ממוינת **בסדר עולה**

על פי כמות הצעדים אותו עשה הכלי. ראשית הדפסו את הצד המנצח ולאחר מכן את הצד המפסיד, בתוך הצדדים על הכלים להיות ממוינים על פי המספר שלהם בסדר עולה. על ההדפסות להיות בפורמט הבא:

```
D1: [(3, 0), (1, 0), (1, 2)]
D13: [(5, 7), (1, 7), (1, 8)]
D2: [(4, 4), (1, 4), (1, 5), (2, 5)]
```

כלומר שחקן מספר 13 של ההגנה (המספרים בתמונה בעמוד הבא), צעד מנקודת ההתחלה שלו, שמאלה אל (1,7) ולאחר מכן המשיך אל (1,8). עבור כל כלי יש לרדת שורה, ורק כלים שזזו במהלך המשחק יופיעו ברשימה.

2. הדפסו את כל הכלים ממוינים **בסדר יורד** על פי כמות האכילות, אם הכמות שווה יש למיין על פי המספר שלהם בסדר עולה, אם המספר שווה יש למיין על פי הקבוצה המנצחת קודם, בפורמט הבא (בכל אכילה רק הכלי שצעד מקבל ++):

```
D4: 4 kills
A12: 4 kills
D2: 3 kills
A24: 3 kills
```

3. הדפסו את כל הכלים ממוינים **בסדר יורד** על פי המרחק המצטבר שצעדו במהלך המשחק, אם המרחק שווה יש למיין על פי מספר הכלי בסדר עולה, אם המספר שווה יש למיין על פי הקבוצה המנצחת קודם. הציגו רק כלים שצעדו לפחות צעד אחד. לדוגמה, כלי שצעד את שני הצעדים האלה:

```
(5, 7), (1, 7), (1, 8)
```

המרחק שצעד הכלי הוא 5. הדפסו על פי הפורמט הבא:

```
D1: 15 squares
K7: 15 squares
D13: 5 squares
```

4. הדפסו את כל המשבצות ממוינות **בסדר עולה** על פי כמות הכלים **השונים** שדרכו עליהם, אם הכמות שווה מיינו על פי הX של המשבצת, אם גם זה שווה מיינו על פי הY. הציגו רק משבצות שלפחות שני כלים שונים דרכו עליהם (חשבו על מבנה נתונים מתאים).

```
(4, 4)1 pieces
(8, 8)2 pieces
(4, 7)4 pieces
```

הערה: חובה להשתמש ב Comparator בכל הסעיפים לעיל, שימו לב לסדר המיון. הניקוד ניתן ע"י בדיקת הפלט ולכן יש להקפיד על כל הפרטים.

GAME RULES:

Hnefatafl is played on a board consisting of an 11 x 11 grid of squares. Accompanying the board, there should be 24 dark pieces and 13 light pieces usually in the form of pawns plus an additional King piece.

0	1	2	3	4	5	6	7	8	9	10	
			1	2	3	4	5				0
					6						1
											2
7					1					8	3
9				2	3	4				10	4
11	12		5	6	7	8	9		13	14	5
15				10	11	12				16	6
17					13					18	7
											8
					19						9
			20	21	22	23	24				10

Hnefatafl is a game of unequal sides and different objectives. The King is placed in the center of the board and the light defending pieces surround him in the pattern shown. The dark pieces are placed in 4 groups in a particular formation at the middle of each edge of the board.

Winning the game

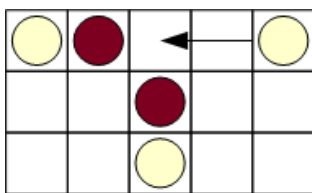
The aim of the defending light-colored side, is to get the King to a corner square of the board.

The aim of the attacking side is to kill the King before he escapes to a corner.

The King is captured by opposing pieces surrounding him on all four sides or, if the King is at the edge of the board, on the 3 sides available. Only the King is allowed into the corner squares – and, of course, such a move wins the game.

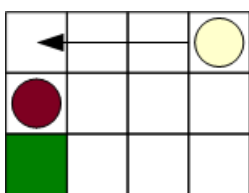
Basic Play

The attacking side moves first. All pieces move like the rook in chess – in a straight line for as many empty squares as the player chooses. Pieces cannot hop over other pieces and cannot move diagonally.

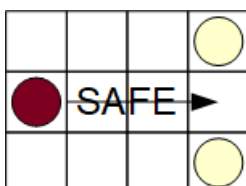


Capturing only happens when a piece is moved so that a single opposing piece ends up trapped between two of the player's pieces. A captured piece is immediately removed from the board. It is possible to capture more than one piece at once.

The King is unarmed and cannot capture, even though a mate pawn started the attack.



A piece that is next to the corner square can be captured by a single opposing piece. The piece is moved so that the opposing piece is trapped between the corner and the piece moved - the corner square acts as the second capturing piece.



A piece can safely move to a square between two of the opponent's pieces without being captured.

בהצלחה!