

עבודה 4 – תיאור בדיקות

בדיקות חלק 1:

מהות	שם בדיקה
בדיקת פעולת החיבור בין שני מספרים ממשיים, ושה- toString מחזיר "+"	TestPart1AddOp1test
בדיקת פעולת החיבור בין שני מספרים שלמים, ושה- toString מחזיר "+"	TestPart1AddOp2test
בדיקת פעולת החיסור בין שני מספרים ממשיים, ושה- toString מחזיר "-"	TestPart1SubtractOp1test
בדיקת פעולת החיסור בין שני מספרים שלמים, ושה- toString מחזיר "-"	TestPart1SubtractOp2test
בדיקת פעולת הכפל בין שני מספרים ממשיים, ושה- toString מחזיר "*"	TestPart1MultiplyOp1test
בדיקת פעולת הכפל בין שני מספרים שלמים, ושה- toString מחזיר "*"	TestPart1MultiplyOp2test
בדיקת פעולת החילוק בין שני מספרים ממשיים, ושה- toString מחזיר "/"	TestPart1DivideOp1test
בדיקת פעולת החילוק בין שני מספרים שלמים, ושה- toString מחזיר "/"	TestPart1DivideOp2test
בדיקת פעולת החזקה בין שני מספרים ממשיים, ושה- toString מחזיר "^"	TestPart1PowOp1test
בדיקת פעולת החזקה בין שני מספרים שלמים, ושה- toString מחזיר "^"	TestPart1PowOp2test
בדיקת פעולת החזקה בין שני מספרים ממשיים ויתכן שהשני שלילי, ושה- toString מחזיר "^"	TestPart1PowOp3test
בדיקת השיטות של ValueToken על ערכים שונים	TestPart1ValueToken1test
בדיקה ש- ExpTokenizer מטפל בכל ה- CalcTokens המוגדרים בחלק זה של העבודה	TestPart1ExpTokenizer1test
בדיקת יכולת הרחבה לפעולות נוספות – פעולת \$ בין שני ValueTokens	TestPart1ExpTokenizerExtendible
בדיקת הביטוי "2 3 +"	TestPart1PostfixCalculator1test
בדיקת הביטוי "3 5 -"	TestPart1PostfixCalculator2test
בדיקת הביטוי "6 2 *"	TestPart1PostfixCalculator3test
בדיקת הביטוי "10 4 /"	TestPart1PostfixCalculator4test
בדיקת הביטוי "2 4 ^"	TestPart1PostfixCalculator5test
בדיקת הביטוי "2 3 + 4 2 - *"	TestPart1PostfixCalculator6test
בדיקת הביטוי "2 3 ^ 4 2 * / 7 -"	TestPart1PostfixCalculator7test
בדיקת הביטוי "2 3 ^ 4 2 * / -7 -"	TestPart1PostfixCalculator8test
בדיקת הביטוי "2 3 +", עם הרצת getCurrentResult() מספר פעמים ולבדוק האם התוצאה המתקבלת היא זהה בכל הפעמים	TestPart1PostfixCalculator9test
בדיקת הביטוי "2 3 ^ 4 2 * / -7 -", עם הרצת getCurrentResult() מספר פעמים ולבדוק האם התוצאה המתקבלת היא זהה בכל הפעמים	TestPart1PostfixCalculator10test

בדיקות חלק 2:

מהות	שם בדיקה
בדיקה של- AddOp ול- SubtractOp ישנה אותה עדיפות	TestPart2BinaryOp1testAddSubtract
בדיקה של- MultiplyOp ול- DivideOp ישנה אותה עדיפות	TestPart2BinaryOp2testMultiplyDivide
בדיקה של- AddOp יש עדיפות קטנה מ- MultiplyOp ול- PowOp יש עדיפות קטנה מ- MultiplyOp	TestPart2BinaryOp3testPrecedenceOrder
בדיקת PeekableStackAsArray: הכנסת איבר אחד וביצוע size	TestPart2PeekableStackAsArray1test1ElementSize
בדיקת PeekableStackAsArray: הכנסת שני איברים וביצוע peek(0) (אמור להחזיר את האיבר בראש המחסנית) וביצוע peek(1) (אמור להחזיר את האיבר השני במחסנית)	TestPart2PeekableStackAsArray1test2ElementsPeek
בדיקת PeekableStackAsArray: הכנסת שני איברים וביצוע peek(2) (אמור לזרוק שגיאת EmptyStackException)	TestPart2PeekableStackAsArray1test2ElementsPeekError
בדיקת PeekableStackAsArray: הכנסת שני איברים והוצאה אחת (אמור להחזיר את האיבר השני שהוכנס)	TestPart2PeekableStackAsArray1test2ElementsPop1
בדיקת PeekableStackAsArray: הכנסת שני איברים, הוצאה אחת, ביצוע clear הכנסה אחת והוצאה אחת (בדיקה שההוצאה האחרונה החזירה את האיבר האחרון שהוכנסו)	TestPart2PeekableStackAsArray1test2ElementsPop1ClearPushPop
בדיקת PeekableStackAsArray: הכנסת שני איברים, הוצאה אחת, ביצוע clear הכנסה אחת ובדיקת גודל (אמור להיות 1)	TestPart2PeekableStackAsArray1test2ElementsPop1ClearPushSize
בדיקת PeekableStackAsArray: הכנסת שני איברים, הוצאה אחת ובדיקת peek(0) (אמור להחזיר את האיבר הראשון שהוכנס)	TestPart2PeekableStackAsArray1test2ElementsPop1Peek
בדיקת PeekableStackAsArray: הכנסת שני איברים, הוצאה אחת ובדיקת peek(1) (אמור לזרוק שגיאת EmptyStackException)	TestPart2PeekableStackAsArray1test2ElementsPop1PeekError
בדיקת PeekableStackAsArray: הכנסת שני איברים, הוצאה אחת, הכנסת 4 איברים, ביצוע clear ובדיקת peek(0) (אמור לזרוק שגיאת EmptyStackException)	TestPart2PeekableStackAsArray1test2ElementsPop1PushManyClearPeek
בדיקת PeekableStackAsArray: הכנסת שני איברים, הוצאה אחת, הכנסת 4 איברים, ביצוע clear וביצוע pop() (אמור לזרוק שגיאת EmptyStackException)	TestPart2PeekableStackAsArray1test2ElementsPop1PushManyClearPop
בדיקת PeekableStackAsArray: הכנסת שני איברים, הוצאה אחת, הכנסת 4 איברים, ביצוע clear וביצוע size() (אמור להחזיר 0)	TestPart2PeekableStackAsArray1test2ElementsPop1PushManyClearSize
בדיקת PeekableStackAsArray: הכנסת שני איברים, הוצאה אחת, הכנסת 4 איברים וביצוע size() (אמור להחזיר 3)	TestPart2PeekableStackAsArray1test2ElementsPop1PushManySize
בדיקת PeekableStackAsArray:	TestPart2PeekableStackAsArray1test2El

הכנסת שני איברים, הוצאה אחת וביצוע size() (אמור להחזיר 1)	ementsPop1Size
בדיקת PeekableStackAsArray: הכנסת שני איברים וביצוע size() (אמור להחזיר 2)	TestPart2PeekableStackAsArray1testElementsSize
בדיקת PeekableStackAsArray: ביצוע clear() וביצוע size() (אמור להחזיר 0)	TestPart2PeekableStackAsArray1testClearEmptyStackSize
בדיקת PeekableStackAsArray: בדיקת peek(0) ו-peek(1) (אמור לזרוק שגיאת EmptyStackException בשני המקרים)	TestPart2PeekableStackAsArray1testEmptyStackPeek
בדיקת PeekableStackAsArray: ביצוע size() (אמור להחזיר 0)	TestPart2PeekableStackAsArray1testEmptyStackSize
בדיקת PeekableStackAsArray: הכנסת 30 איברים ובדיקה שאילו האיברים שמוחזרים בהפעלת peek לפי הסדר המתאים	TestPart2PeekableStackAsArray1testManyElementsPeek
בדיקת PeekableStackAsArray: הכנסת 30 איברים ובדיקת peek(30) (אמור לזרוק שגיאת EmptyStackException)	TestPart2PeekableStackAsArray1testManyElementsPeekError
בדיקת PeekableStackAsArray: הכנסת 30 איברים ובדיקת 25 ה- pop() הראשונים	TestPart2PeekableStackAsArray1testManyElementsPop
בדיקת PeekableStackAsArray: הכנסת 30 איברים, ביצוע 25 pop(), ביצוע clear(), הכנסה אחת ובדיקה ש-peek(0) מחזיר את האיבר האחרון שהכנסנו	TestPart2PeekableStackAsArray1testManyElementsPopClearPushPeek
בדיקת PeekableStackAsArray: הכנסת 30 איברים, ביצוע 25 pop(), ביצוע clear(), הכנסה אחת ובדיקה ש-peek() מחזיר את האיבר האחרון שהכנסנו	TestPart2PeekableStackAsArray1testManyElementsPopClearPushPop
בדיקת PeekableStackAsArray: הכנסת 30 איברים, ביצוע 25 pop(), ביצוע clear() ובדיקה ש- size() מחזיר 0	TestPart2PeekableStackAsArray1testManyElementsPopClearSize
בדיקת PeekableStackAsArray: הכנסת 30 איברים, ביצוע 25 pop() ובדיקה ש- size() מחזיר 5	TestPart2PeekableStackAsArray1testManyElementsPopSize
בדיקת PeekableStackAsArray: הכנסת 30 איברים ובדיקה ש- size() מחזיר 30	TestPart2PeekableStackAsArray1testManyElementsSize
בדיקת PeekableStackAsArray: הכנסת איברים שהם Integer ו- Double ובדיקה של size()	TestPart2PeekableStackAsArray2testIntegersPushSize
בדיקת PeekableStackAsArray: הכנסת איברים שהם Integer ו- Double ובדיקה ש-peek() של כל האיברים מחזירים את האיברים בסדר הפוך	TestPart2PeekableStackAsArray2testIntegersPushPop
בדיקת PeekableStackAsArray: הכנסת איברים שהם Integer ו- Double, הוצאת כל האיברים, ובדיקה ש-peek() זורק שגיאת EmptyStackException	TestPart2PeekableStackAsArray2testIntegersPushPopError
בדיקת PeekableStackAsArray: הכנסת איברים שהם Integer ו- Double ובדיקה ש-	TestPart2PeekableStackAsArray2testIntegersPushPeek

peek(i) של כל אחד מהאיברים מחזיר את האיבר הנכון	
בדיקת PeekableStackAsArray: הכנסת 7 איברים שהם Integer ו- Double ובדיקה ש- peek(7) זורק שגיאת EmptyStackException	TestPart2PeekableStackAsArray2testIntegersPushPeekError
בדיקה ש- ExpTokenizer תומך בביטוי "(" (countTokens הוא 1)	TestPart2ExpTokenizer1testOpenBracket1
בדיקה ש- ExpTokenizer תומך בביטוי "(" (הביטוי של nextElement הראשון הוא OpenBracket)	TestPart2ExpTokenizer1testOpenBracket2
בדיקה ש- ExpTokenizer תומך בביטוי ")" (countTokens הוא 1)	TestPart2ExpTokenizer1testCloseBracket1
בדיקה ש- ExpTokenizer תומך בביטוי ")" (הביטוי של nextElement הראשון הוא CloseBracket)	TestPart2ExpTokenizer1testCloseBracket2
בדיקה ש- ExpTokenizer תומך בביטוי "()" (countTokens הוא 2)	TestPart2ExpTokenizer1testOpenCloseBracket1
בדיקה ש- ExpTokenizer תומך בביטוי עם מספר אקראי של סוגריים ומחזיר אותם ע"פ הסדר	TestPart2ExpTokenizer1testManyOpenCloseBracket
בדיקת הביטוי "2 + 3"	TestPart2InfixCalculator1test
בדיקת הביטוי "3 - 5"	TestPart2InfixCalculator2test
בדיקת הביטוי "6 * 2"	TestPart2InfixCalculator3test
בדיקת הביטוי "10 / 4"	TestPart2InfixCalculator4test
בדיקת הביטוי "2 ^ 4"	TestPart2InfixCalculator5test
בדיקת הביטוי "2 + 3 * 4 - 2"	TestPart2InfixCalculator6test1
בדיקת הביטוי "(2 + 3) * (4 - 2)"	TestPart2InfixCalculator6test2
בדיקת הביטוי "2 ^ 3 / 4 * 2 - 7"	TestPart2InfixCalculator7test1
בדיקת הביטוי "((2 ^ 3) / (4 * 2)) - 7"	TestPart2InfixCalculator7test2
בדיקת הביטוי "2 ^ 3 / 4 * 2 - 7", ושה- getCurrentResult() מחזיר את אותה התוצאה כמה פעמים	TestPart2InfixCalculator8test1
בדיקת הביטוי "((2 ^ 3) / (4 * 2)) - 7", ושה- getCurrentResult() מחזיר את אותה התוצאה כמה פעמים	TestPart2InfixCalculator8test2
בדיקת הביטוי "2 + 3", ושה- getCurrentResult() מחזיר את אותה התוצאה כמה פעמים	TestPart2InfixCalculator9test
בדיקת הביטוי "2 ^ 3 / 4 * 2 - 7", ושה- getCurrentResult() מחזיר את אותה התוצאה כמה פעמים	TestPart2InfixCalculator10test1
בדיקת הביטוי "((2 ^ 3) / (4 * 2)) - 7", ושה- getCurrentResult() מחזיר את אותה התוצאה כמה פעמים	TestPart2InfixCalculator10test2
בדיקה שאין במחלקה PeekableStackAsArray מערכים או שדה מסוג Stack	TestPeekableStackCheckFields
בדיקה ש- PeekableStackAsArray מרחיב את StackAsArray	TestPeekableStackExtendsStack

בדיקות Tester:

מהות	שם בדיקה
לפחות 20 בדיקות ראשונות עברו בהצלחה	TestMoreThan20Tests
לפחות 35 בדיקות ראשונות עברו בהצלחה	TestMoreThan35Tests
לפחות 50 בדיקות ראשונות עברו בהצלחה	TestMoreThan50Tests

בדיקות Bonus:

מהות	שם בדיקה
בדיקת הביטוי "2 % 5" (פעולת ה- % לא חוקית)	TestBonusParseException1testParseException1
בדיקת הביטוי "5" (הסוגריים לא חוקיים)	TestBonusParseException1testParseException2
בדיקת הביטוי "5" (הסוגריים לא חוקיים)	TestBonusParseException1testParseException3
בדיקת הביטוי "7.0.1" (token לא חוקי)	TestBonusParseException1testParseException4
בדיקת הביטוי "+ 5" (פעולת + צריכה לקבל עוד פרמטר)	TestBonusParseException1testParseException5
בדיקת הביטוי "+ + 7" (פעולת + צריכה לקבל עוד פרמטר מספרי)	TestBonusParseException1testParseException6
בדיקת הביטוי "+ + 5 7" (פעולת + צריכה לקבל עוד פרמטר מספרי)	TestBonusParseException1testParseException7
בדיקת הביטוי "+ 7" (פעולת + צריכה לקבל עוד פרמטר)	TestBonusParseException1testParseException8
בדיקת הביטוי "5 7" (לאחר מספר יש עוד מספר)	TestBonusParseException1testParseException9
בדיקת הביטוי "+" (פעולת + צריכה לקבל שני פרמטרים)	TestBonusParseException1testParseException10
בדיקת הביטוי "+ 5" (אמור לזרוק ParseException), ואז ביצוע שערך של "7 + 2" על אותו מחשבון, וקבלת תשובה תקינה	TestBonusParseException1testParseExceptionCalcAfterException