מגישים:

אופיר קרנגל 302621305

רומן אברמזון 306359001

Contents

[שאלה 1 2](#_Toc529896235)

[שאלה 2 3](#_Toc529896236)

[שאלה 3 4](#_Toc529896237)

[שאלה 4 5](#_Toc529896238)

[חלק א 5](#_Toc529896239)

[חלק ב 5](#_Toc529896240)

# שאלה 1

Coding Style guidelines

1. Introduction

The scope of this document is to describe the coding style guidelines which will be used throughout the course by our team, the purpose of these guidelines is to ensure readability and easy maintenance of the evaluated designs.

2. Naming conventions

* variables, methods, classes, arrays etc should be given short names that describe correctly the required behavior
* Constants and parameters should be in capital
* Delimit separate words with underscore “\_”
  + For example: current\_value
* Avoid using camel notation
  + For example: currentValue

3. Documentation

* Use comments when needed
* Try to explain corner cases in details
* Keep code readable as possible

4. Indentation

* Use tabs instead of whitespace
* A code within a block should have the same indentation
  + For example:
  + case {

A

B

}

* Block should be separated with braces, even if the block consist of single line
* Use empty lines to separate between different functionally

# שאלה 2

פלט של התוכנית לקובץ comment.txt:

/\*\*

\* RatNum represents an immutable rational number.

\* It includes all of the elements of the set of rationals, as well

\* as the special "NaN" (not-a-number) element that results from

\* division by zero.

\*

\* The "NaN" element is special in many ways. Any arithmetic

\* operation (such as addition) involving "NaN" will return "NaN".

\* With respect to comparison operations, such as less-than, "NaN" is

\* considered equal to itself, and larger than all other rationals.

\*

\* Examples of RatNums include "-1/13", "53/7", "4", "NaN", and "0".

\*/

// Abstraction Function:

// A RatNum r is NaN if r.denom = 0, (r.numer / r.denom)

// otherwise.

// Representation invariant for every RatNum r:

// (r.denom >= 0) &&

// (r.denom > 0 ==> there does not exist integer i > 1 such that

// r.numer mod i = 0 and r.denom mod i = 0)

// (in other words: the denom. is always non-negative and if the

// denom. is non-zero, the fraction represented is in reduced form.)

/\*\*

\* Constructs a new RatNum.

\* @effects Constructs a new RatNum = n.

\*/

/\*\*

\* Constructs a new RatNum.

\* @effects If d = 0, constructs a new RatNum = NaN. Else

\* constructs a new RatNum = (n / d).

\*/

// â€¦

/\*\*

\* Checks to see if the representation invariant is being violated.

\* @throws AssertionError if representation invariant is violated.

\*/

/\*\*

\* Returns the greatest common divisor of 'a' and 'b'.

\* @requires b != 0

\* @return d such that a % d = 0 and b % d = 0

\*/

// â€¦

/\*\*

\* @requires arg != null

\* @return a new RatNum equal to (this / arg).

\* If arg is zero, or if either argument is NaN, then returns NaN.

\*/

// (a/b) / (x/y) = ay/bx

//â€¦

# שאלה 3

1. פלט תוכנית ההרצה

Get total volume

0.0

Add ball0(1.200000)

true

Add ball1(2.300000)

true

Add ball2(3.500000)

true

Get total volume

7.0

Add ball1

false

Remove ball1

true

Remove ball1

false

Does container contain ball0?

true

Does container contain ball1?

false

Does container contain ball2?

true

Get total volume

4.7

Clear container

Get total volume

0.0

1. BallContainer0 מייצג את דרך א שסוכמת את הסכום בכל הוספת כדור חדש או מחסירה בכל הסרה של כדור קיים. BallContainer1 מייצג דרך ב בה סוכמים פעם אחת את כל הכדורים ע"י מעבר בלולאה על מערך הכדורים.

יתרונות גישה א:

* הסכימה ידועה בכל רגע נתון ולא מצריכה חישוב מחדש בכל פעם שניקרא לפונקציה (תשובה חוזרת במיידי)
* יעילה בעיקר כשיש הרבה כדורים או כאשר קוראים לפונקציה הרבה פעמים.

יתרונות גישה ב:

* לא מצריכה משתנה עזר במחלקה.
* יותר קלה לתחזוק כיוון והקוד מרוכז במקום אחד ולא במספר מקומות שונים בקוד. הוספה של פונקציה חדשה לא תצריך התייחסות למשתנה העזר.
* יעילה בעיקר כשיש מעט כדורים או כאשר קוראים לפונקציה מעט פעמים.

ג.

1. כן. השינוי במפרט יהיה הסרת ערך ה return מכיוון שתמיד יגיע כדור לא null.
2. כן. השינוי שיעשה הוא הסרת בדיקת האם הכדור קיים כיוון שזה באחריות המשתמש ושינוי הפונקציה ל void.
3. המפרט חלש יותר מכיוון שהוא תומך בפחות קלטים.

# שאלה 4

## חלק א

תשובה א: לכל מפרט יש נקודת חוזק חזקה על מישנהו ולכן לא ניתן לקבוע יחס חוזק בין שני המפרטים.

## חלק ב

1. תשובה ד: מפרט S1 חזק יותר כיוון שהוא תומך בכל קלט אפשרי וממש את כל מה שהתבקש מ S2,S4 ולכן הוא מקיים את S2,S4 אך מממש חלקית ממפרט S3 ולכן לא מקיים את S3.
2. תשובה ו: לא מקיים את S1 כי הוא לא עונה על כל קלט אפשרי. לא מקיים את S3 כי הוא מממש אותו חלקית. לא מקיים את S4 כי הוא לא עונה על דרישה שהמערך יהיה ממויין.
3. תשובה ג: לא מקיים את S1,S2 כיוון שלא מחזיר -1 בשגיאה. מקיים את S4 מבחינת דרישות קלט ומימוש.
4. תשובה ו: לא מקיים את S1,S3 בקלט. לא מקיים את S2 במימוש.