תרגול קורס גאווה מתקדם – מערכת לניהול פניות

**תרגול 1**

במהלך הקורס נממש מערכת לפניות לשירות לקוחות.

בשיעור זה נממש יצירת פניות שונות וטיפול בהם, כל פנייה תטופל על בנפרד.

תוכן

[***חלק א – יצירת המחלקות*** 1](#_Toc189562556)

[***חלק ב – טיפול בנפרד לכל פניה*** 1](#_Toc189562557)

[***חלק ג – תיעדוף סוגי הפניה*** 3](#_Toc189562558)

[***חלק ד – רשות - הדמית זמן ריצה של הטיפול בפניה*** 3](#_Toc189562559)

[***חלק ה – ניהול יצירת והרצת הטיפול בפניות ע"י תור*** 4](#_Toc189562560)

[***חלק ו – בונוס*** 4](#_Toc189562561)

# חלק א – יצירת המחלקות

1. צרי new package – Data
2. צרי וממשי בתוכו את המחלקות הבאות:

* Inquiry – מחלקה ראשית המייצגת פניה.
* Question – מחלקה יורשת של פניה לשאילות
* Request – מחלקה יורשת של פניה לבקשות.
* Complaint – מחלקה יורשת של פניה לתלונות

1. הוסיפי במחלקה Inquiry משתנה סטטי למספור קוד הפניה:

**static** Integer *nextCodeVal* = 0;

1. ממשי שלכל המחלקות יהיו את המשתנים:

* Integer code – קוד מזהה יחודי לכל פניה.
* String description – תיאור הפניה
* LocalDateTime creationDate – תאריך ושעת יצירת הפניה.

1. בכל המחלקות ממשי פונקציה fillDataByUser שמאתחלת את משתני המחלקה ע"י קלט מהמשתמש או הצבת נתונים.
2. את המשתנה הבא הוסיפי רק ל Complaint

assignedBranch – שם הסניף אליו מקושרת התלונה.

1. בכל מחלקה ממשי פונקציה handling הכותבת הודעה שהמערכת מטפלת בפניה <סוג הפניה> מס' <קוד הפניה>.

# חלק ב – טיפול בנפרד לכל פניה

1. צרי new package business
2. הוסיפי לו מחלקה *InquiryHandling* – המחלקה מריצה טיפול בנפרד לכל פניה. המחלקה מכילה משתנה

***private*** *Inquiry currentInquiry;*

1. ממשי במחלקה פונקציה createInquiry ליצירת הפניה ושמירתה במחלקה:

הפונקציה מבקשת מהמשתמש להקיש 1 – לשאלה, 2 – לבקשה, 3 – לתלונה. יוצרת פניה מתאימה ומציבה אותה ב *currentInquiry.*

1. ממשי הפעלה של הטיפול בפניה (*currentInquiry.handling()*) עבור הפניה שנוצרה כך שהיא תורץ בנפרד עבור כל פניה שנוצרת במערכת.
2. בכל טיפול בפניה בצעי השהיה של 5 שניות (sleep)
3. **סיכום:**

הריצי את קטעי הקוד הבאים בmain ב2 ריצות נפרדות first running and Second running:

*InquiryHandling handling1 =* ***new*** *InquiryHandling();*

*Thread handling2 =* ***new*** *InquiryHandling();*

*InquiryHandling handling3 =* ***new*** *InquiryHandling();*

*InquiryHandling handling4 =* ***new*** *InquiryHandling();*

*handling1. createInquiry();*

*handling2.createInquiry();*

*handling3.createInquiry();*

*handling4.createInquiry();*

*// first running*

*// handling1.start();*

*// handling2.start();*

*// handling3.start();*

*// handling4.start();*

*// second running*

*handling1.run();*

*handling2.run();*

*handling3.run();*

*handling4.run();*

מה יודפס בקטע ההרצה הראשון?יודפס תיאור הפניה והקוד שלה לא לפי סדר \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

מה יודפס בקטע ההרצה השני? יודפס תיאור הפניה והקוד שלה לפי סדר \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

הסבירי את הסיבה להבדל בין 2 הריצות:

\_בראשון- הופעלה הפונקציה start שמריצה את כולם במקביל, בשניה-הופעלה הפונקציה run שמריצה את הפונקציות לפי סדר כלומר, את הראשון עד שנגמר ורק אחכ את השני וכן הלאה.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

שני את קוד כך שלפניה מסוג בקשה תהיה השהיה של 3 שניות.

מה ואיפה שינית בקוד? מה השתנה בהדפסה?

\_\_\_\_ **public** **void** run() {

**if**(currentInquiry **instanceof** Request)

{

**try** {

Thread.*currentThread*().*sleep*(3000);

} **catch** (Exception e) {

}

}

**else**

{

**try** {

Thread.*currentThread*().*sleep*(5000);

} **catch** (Exception e) {

}

}

currentInquiry.handling();

}

הוא מדפיס קודם את הפניות מסוג בקשה \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# חלק ג – תיעדוף סוגי הפניה

שני את הקוד כך שפניות מסוג שאילה יקבלו טיפול בעדיפות גבוהה.

הריצי את קטע הmain מספר פעמים וודאי שפניות מסוג שאילות מקבלות מענה ראשון.

שימי לב איפה לממש את תיעדוף הפניה, האם לפני ההשהיה או לאחריה...

# חלק ד – רשות - הדמית זמן ריצה של הטיפול בפניה

הערכת זמני הטיפול בפניות הן:

* שאילה: 1-5 שניות
* בקשה 10-15 שניות
* תלונה 20-40 שניות

יש לשנות את קוד הטיפול בפניה כך שיפעיל השהיה בטווח הטיפול בפניה.

בשביל הגרלת זמן יש להשתמש במחלקה Random.

1. הגרילי מספר המייצג את זמן הטיפול בפניה בטווח של הערכות זמני הטיפול.
2. הדפיסי הודעה של סוג ומספר הפניה ומשך זמן הטיפול המשוער.
3. בצעי השהיה לטיפול בפניה למשך זמן הטיפול המשוער.
4. במידה וזמן הטיפול בפניה גדול מ5 שניות ויש יותר מ10 תהליכים שרצים במערכת שחררי את המערכת לרוץ על תהליכים נוספים.

שימי לב איפה לממש חלק זה – האם לפני ההשהיה או לאחריה

1. כתבי קוד ב main שיוצר 15 פניות ומריץ אותם במקביל.
2. נתחי את התוצאות.

פונקציות שיש להשתמש בהם בתרגיל:

*Thread.currentThread().sleep(x\*1000);*

*Thread.currentThread().yield();*

*Thread.currentThread().activeCount();*

*Random rand =* ***new*** *Random();*

***int*** *estimationTime = rand.nextInt(10, 20);*

1. **סיכום:**

מה ההבדל בין sleep ל yield?

\_sleep מרדים את התהליך וyield מפסיק את התהליך הנוכחי ומריץ את שאר התהליכים \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# חלק ה – ניהול יצירת והרצת הטיפול בפניות ע"י תור

1. הוסיפי ל business package מחלקה InquiryManager – המחלקה מנהלת את היצירה והטיפול בפניות.
2. נהלי את הטיפול בפניות לפי תור – פניה ראשונה שנוצרה מטופלת ראשונה.
3. כתבי פונקציה inquiryCreation הפונקציה יוצרת פניות לטיפול (*InquiryHandling)* ע"י קלט מהמשתמש.

לסיום יש להקיש exit

את כל הפניות לטיפול שנוצרו יש לשמור בתור.

1. כתבי פונקציה processInquiryManager() המנהלת את הפעלת הטיפול בפניה.

שימי לב להפעיל את הטיפול בפניות לפי סדר התור שהן נוצרו.

כל טיפול בפניה צריך לרוץ בThread נפרד.

1. **סיכום:**

הפעילי את הקוד הבא בmain ובדקי את המערכת שכתבת.

*InquiryManager inquiryManager = new InquiryManager();*

*inquiryManager.inquiryCreation();*

*inquiryManager.processInquiryManager();*

# חלק ו – בונוס

שפרי את המערכת כך שתוכלי להמשיך וליצור פניות נוספות בזמן שהמערכת מטפלת בפניות הקיימות.