קורס גאווה מתקדם – מערכת לניהול פניות

**תרגול 4**

בשיעור זה נבנה תשתית קוד לניהול ארכיטקטורת client-server, עבודה עם אובייקטים ופרוטוקול מוסכם מראש. תמיכה במספר לקוחות בו זמנית.

תוכן

[***אפיון דרישות תשתית שרת לקוח*** 1](#_Toc196149916)

[***חלק א – תכנון Hight Level Design*** 1](#_Toc196149917)

[***חלק ב – כתיבת Design*** 1](#_Toc196149918)

[***חלק ג – פיתוח*** 2](#_Toc196149919)

[***למתקדמות – Pull Request*** 2](#_Toc196149920)

# אפיון דרישות תשתית שרת לקוח

**מידע כללי**

תשתית שרת לקוח שלנו מבוססת על תבנית של שליחת אובייקטים.

שרת מאזין להתחברות לקוח, לכל לקוח שמתחבר מתבצע תהליך קריאת הבקשה, עיבוד ושליחת תשובה.

השרת פנוי לתמוך במספר לקוחות בו זמנית.

לקוח יכול לבצע התחברות לשרת שליחת בקשה וקבלת תשובה.

**הדרישות**

צד שרת server:

1. האזנה לפורט קבוע (למשל 12345) באמצעות ServerSocket.
2. קבלת חיבורים מרובים:
   * כל לקוח יקבל Thread או Runnable עצמאי.

אופציונלית: שימוש ב־ExecutorService.

* + עיבוד בקשות מסוג Request:
    - כל בקשה היא אובייקט הנשלח מהלקוח.
    - בקשה כוללת action ו־params.
  + שליחת תשובה מסוג Response:
    - תשובה יכולה להכיל:
    - סטטוס (success, error)
    - הודעה טקסטואלית
    - אובייקט תוצאה – מכל טיפוס (Object result)

1. ניהול חיבור – שליחת תשובה וסגירת החיבור.

צד לקוח client:

1. חיבור לשרת בכתובת ופורט נתונים (למשל localhost:12345).
2. בניית אובייקט בקשה
3. שליחת בקשה לשרת דרך ObjectOutputStream.
4. קבלת תגובה מהשרת:
   * קריאת Response עם שדות:
   * success האם הבקשה הצליחה
   * message הודעה אנושית
   * result תוצאה מסוג כלשהו (מספר, מחרוזת, רשימה, אובייקט מורכב וכו')
5. שימוש בתוצאה – הדפסתה.

מערכת ניהול פניות תאפשר לבצע את הפעולות הבאות ע"י לקוח ממחשב מרוחק:

1. הוספת פניה למאגר.
2. אחזרת כל הפניות הקיימות במאגר.

**הכוונה טכנית**

* עבודה עם ObjectInputStream ו־ObjectOutputStream.
* כל מחלקות האובייקטים חייבות לממש Serializable
* ניהול החיבורים בשרת עם Threads
* שמירה על מבנה קוד נקי ומופרד:
* מחלקות Request / Response
* מחלקת שרת
* מחלקת לקוח

# חלק א – תכנון Hight Level Design

קראו ביחד את הדרישות לעיל של אפיון תשתית שרת לקוח.

תכננו ביחד את הקוד ברמה הכללית:

packages, classes and the member of class, main process and relevant functions

תכננו את המחלקות שאתן צריכות ברמה הכללית, ומה הן מכילות. תנו שמות למחלקות ולמשתנים הראשיים, עברו על התהליכים באפיון ולכל תהליך תכננו אלו פונקציות ובאילו מחלקות יממשו את התהליך.

חלקו את האפיון לתת משימות וחלקו ביניכן את העבודה. תכננו חלוקת עבודה ככה שתוכלו לעבוד במקביל ככל האפשר.

# חלק ב – כתיבת Design

כל אחת ממכן תכתוב מסמך WORD של תכנון מפורט מה היא הולכת לפתח.

לכל תת משימה כתבי פירוט של המחלקות והמשתנים, שמות המשתנים, שמות הפונקציות והמטרה של כל פונקציה.

# חלק ג – פיתוח

עבדו יחד על אותו הrepository .

ממשו כל אחת את חלק הפיתוח שלה במקביל.

הכניסו את הקוד.

בצעו בדיקות על כל התהליך ופתרו באגים בעת הצורך.

**שימו לב !! לעבוד מסודר וברור, שמות מחלקות ומשתנים ברורים ש"מספרים" על עצמם ללא צורך בהערות. חלוקה לתתי משימות. בראנצים מסודרים עם שמות ברורים לכל תת משימה. אבחנה בשמות הבראנשים לפיתוח/לתיקוני באגים.**

אם עבדתן ותכננתן נכון החלק הזה צריך להיות מהיר ועם מינימום באגים ! :)

# למתקדמות – Pull Request

הכניסו קוד ע"י יצירה של pull request.

כאשר את מסיימת לקודד צרי PR ושילחי אותו לאישור של חברתך.

רק לאחר שהיא עברה על הקוד ואישרה הכניסי את הקוד לבראנץ הראשי הmaster.