**פרויקט גמר**

**תכנות מקבילי ומבוזר**

נושא הפרויקט: בחירות לראשות הממשלה

שם הסטודנט: תהילה אברהמי

מספר זהות: 213289754

מנחה: המורה שרי שנהב

**תוכן עניינים**

מבוא............................................................................... עמ' 3

מדריך למשתמש............................................................... עמ' 4

בעיות שהסנכרון פותר........................................................ עמ' 7

פירוט מחלקות.................................................................. עמ' 7

קטעי קוד......................................................................... עמ' 9

**מבוא**

**תכנות מקבילי ומבוזר – מה זה אומר?**

תכנות מקבילי – פירוק האלגוריתם לחלקים קטנים יותר לשם התאמתו לריצה על מספר מעבדים במקביל, שהם חלק מאותו מחשב, לשם האצת הריצה.

חישוב מבוזר – להבדיל מחישוב מקבילי, בחישוב מבוזר המשימה מחולקת בין מחשבים עצמאיים המקושרים באמצעות רשת.

תכנות באמצעות תהליכונים – ביצוע מספר משימות במקביל במסגרת אותו תהליך, תחת מרחב כתובות אחד.

כדי ליישם תכנות מקבילי יש לפרק את הבעיה לגורמים קטנים יותר. במצב האידיאלי החלוקה תהיה סימטרית - כלומר כל תת־בעיה זהה לבעיה הגדולה, פרט לכך שהיא פועלת על אוסף נתונים קטן יותר - חלק מאוסף הנתונים של הבעיה הגדולה. כך ניתן להפעיל את האלגוריתם על מספר משתנה של מעבדים ללא צורך לשנות אותו, תוך שיפור היעילות באופן יחסי למספר המעבדים (אך לא ליניארי).

**מהו סינכרון?**

במערכות מקביליות קיימת בעיה של אי ודאות באשר לסדר הפעולות, היות שכל מעבד מבצע את התוכנית בקצב משלו. הבעיה עלולה ליצור טעויות אלגוריתמיות הנובעות מחוסר תיאום בין המעבדים.

הפרויקט שלי מבטא מערכת בחירות לראשות ממשלתית אשר בה ניתן להצביע בצורה מהירה וקלה בלי לצאת מהבית וללכת לקלפי. המקביליות מתבטאת בכך ששני משתמשים יכולים לבחור בו זמנית והמספר יתעדכן כראוי. כל הקטע של בחירת מפלגה נמצאת בתוך מנעול וזאת כדי לאפשר את המקביליות. לאחר מכן המנעול משתחרר ועובר למשתמש הבא.

כלי השימוש שבו השתמשתי הוא synchronized.

**מדריך למשתמש**

תחילה הרץ את השרת – מחלקת Elections\_server.

לאחר מכן הרץ את מחלקת Home - יפתח חלון של דף הבית ובו אפשרות להצבעה ולצפייה בתוצאות הבחירות לרגע זה.



בלחיצה על בחירות יפתח חלון להצבעה – בחירת המפלגה הרצויה.



בשלב זה ניתן לבחור את המפלגה הרצויה.



בלחיצה על מפלגה מסוימת יתעדכן מספר הקולות למפלגה זו.

לאחר בחירת מפלגה נסגרת האפשרות לבחירה נוספת זאת באמצעות הפקודה setEnabled.

כעת ניתן לצפות בתוצאות הבחירות ברגע זה.



מסך זה מתעדכן בכל הצבעה נוספת.

דוגמא למצב הבחירות ברגע מסוים:



**בעיות שהסנכרון פותר**

משתמש מסוים נכנס למערכת ובוחר מפלגה. באותו זמן בדיוק נכנס משתמש נוסף למערכת וגם הוא בוחר מפלגה. במצב זה מופעל הסינכרון ומעדכן כראוי את מספר הקולות למפלגה.

**פירוט מחלקות**

**צד שרת - Elections\_server**

מחלקת ConectionClient

מחלקה זו מטפלת בהעברת מידע מה - server ל – client.

|  |  |
| --- | --- |
| **פירוט** | **פונקציה** |
| פונקציה זו אחרית על התקשורת בין השרת והלקוח | **public ConectionClient(Socket socket)** |
| פונקציה זו מציגה את מצב המפלגות | **public void ballot(String read)** |
| פונקציה זו מאזינה באופן תמידי לבקשות הקליינט. אם הקליינט לחץ על בחירות מופעלת הפונקציה ballot, אם נבחר צפייה בתוצאות הבחירות – נשלח ללקוח מערך מעודכן המכיל את שם המפלגה ואת מספר הקולות בה לרגע זה. | **public void run()** |

מחלקת Elections\_server

מחלקה זו אחראית על הפעלת התוכנית.

|  |  |
| --- | --- |
| **פירוט** | **פונקציה** |
| הפעלת השרת | **public static void main(String[] args)** |
| פעולה בונה המעדכנת את מערך המפלגות - יוצרת מופעים למפלגות, בפעולה זו גם נוצר השרת וההתקשרות ללקוח. | **public Elections\_server()** |
| פונקציה זו מעדכנת את מספר הקולות במפלגה הנבחרת | **synchronized public static int elections(String read)** |

מחלקת Party

מחלקה זו מכילה את תכונות המפלגה – שם ומספר קולות.

**צד לקוח - Elections\_client**

מחלקת Ballot

זוהי מחלקה גרפית המציגה את המפלגות בבחירות. במחלקה זו מופעלים הכפתורים לבחירה. בכל לחיצה על כפתור נשלחת המפלגה הנבחרת לפונקציה המבצעת את הבחירה.

|  |  |
| --- | --- |
| **פירוט** | **פונקציה** |
| פונקציה להופעת alert – חלונית המאשרת את הבחירה. | **public void alert()** |
| פונקציה המכבה את כל הכפתורים לאחר בחירת מפלגה. | **public void off()** |

מחלקת Elections\_client

מחלקה זו אחראית ליצירת הקשר עם השרת.

|  |  |
| --- | --- |
| **פירוט** | **פונקציה** |
| הפונקציה אחראית על יצירת קשר עם השרת בפועל | **public Elections\_client()** |
| כתיבת הלקוח לשרת | **public void ToServer(String c)** |
| קריאה מהשרת | **public String readFromServer()** |
| יצירת לקוח חדש | **public static void main(String[] args)** |

מחלקת Home

מחלקה גרפית המציגה את דף הבית. במחלקה מופעלים הכפתורים המוצגים בדף הבית.

מחלקת Results

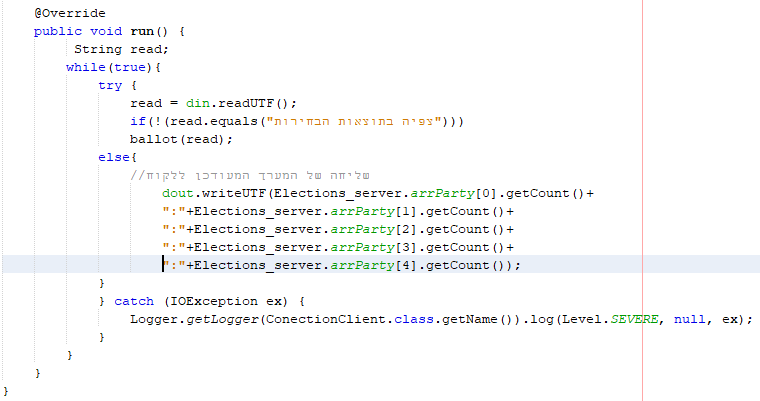
מחלקה גרפית המציגה את תוצאות הבחירות המעודכנות לרגע מסוים. במחלקה קיים Thread אנונימי היוצר את המקביליות.

**קטעי קוד**

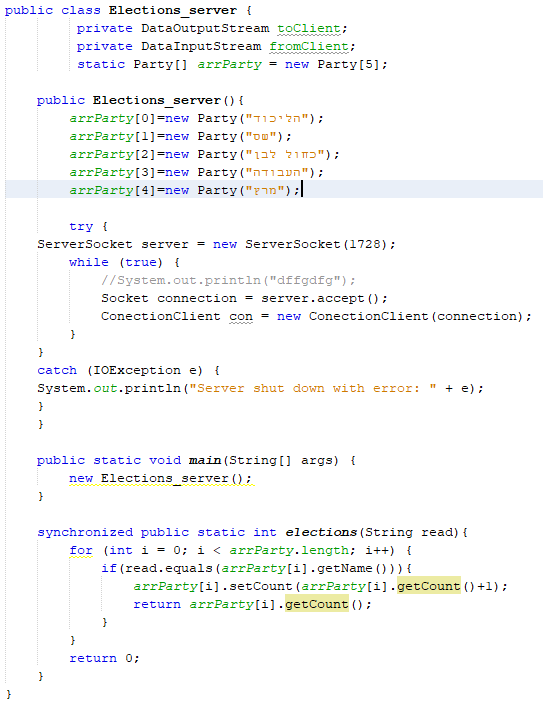
**צד שרת - Elections\_server**

מחלקת ConectionClient

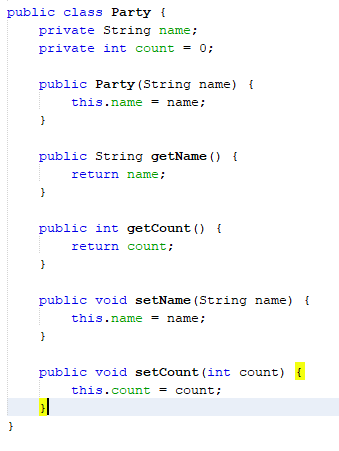
****

****

מחלקת Elections\_server



מחלקת Party



**צד לקוח - Elections\_client**

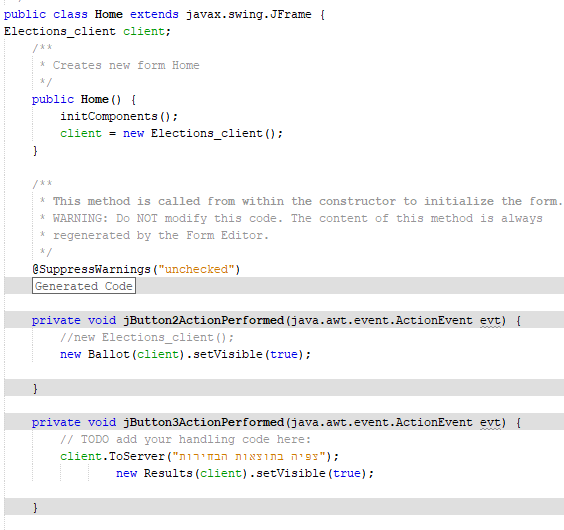
מחלקת Ballot



מחלקת Elections\_client



מחלקת Home



מחלקת Results

