

+

•

0

# מערכת ניהול ספרייה - מבוא וארכיטקטורה

- קורס: תכנות בסביבת האינטרנט
- מרצה: ניסים ברמי
- שלום, אנחנו סהר והילה וזה הפרויקט שלנו לקורס תכנות בסביבת האינטרנט.
- הפרויקט שלנו נקרא מערכת ניהול ספרייה. המערכת שלנו מתוכננת בארכיטקטורת שרת-לקוח.
- צד לקוח: ממשק משתמש ב-JavaFX המאפשר למשתמשים לבצע פעולות.
- צד שרת: אפליקציה חזקה שמטפלת בבקשות הלקוח
- אחסון נתונים: אחסון הנתונים בצורה מאובטחת - באמצעות הקבצים books.txt, customers.txt

+

•

0

# מחלקות חשובות בפרויקט

בפרויקט שלנו יש כמה מחלקות חשובות:

- מודל אלגוריתם: יישום אלגוריתמי מטמון כמו LRU ו-Random Cache כדי לייעל את אחזור הנתונים. (קבצים: LRUCache.java, RandomCache.java)
- מחלקת **Server**: מנהלת ומטפלת בבקשות לקוח.
- מחלקת **Service**: כוללת את הלוגיקה העסקית ופועלת על מודלי נתונים.
- חיבור **Client-service**: מנהל את החיבור בין הלקוח לשרת.
- בדיקת אלגוריתם באמצעות מבחני יחידה ומבחני אינטגרציה כדי לוודא את נכונותו.

+

•

0

# מודול אלגוריתם ובדיקות

- במודול האלגוריתם, יישמנו את האלגוריתמים LRU ו-Random. אם נסתכל במחלקת **LRUCache**, היא אחראית על יישום אלגוריתם המטמון LRU כדי **לייעל** את אחזור הנתונים.
- מחלקת **RandomCache** אחראית על יישום אלגוריתם מטמון אקראי לאחזור נתונים.
- בדקנו את המערכת באמצעות מבחני יחידה ומבחני אינטגרציה כדי לוודא את נכונותה.
- במחלקת **AlgoCacheTest** יש לנו מבחנים לאלגוריתמי המטמון.

# מחלקות שרת ושירות

## חיבור לקוח - שרת

- בחלקים B,C של הפרויקט ישנם מספר קבצים חשובים:
- `BookController.java` מחלקה שאחראית על ניהול הפעולות הקשורות לספרים, כולל הוספה, עדכון ומחיקה של ספרים.
- `HandleRequest.java` מחלקה שמטפלת בבקשות שמתקבלות מהלקוח ומפנה אותן לפונקציות המתאימות.
- `Server.java` מחלקה שמנהלת את השרת ומטפלת בבקשות לקוח נכנסות.
- `ServerDriver.java` מחלקה שמפעילה את השרת ומוודאת שהשרת פועל כראוי.
- `App.java` מחלקת `main` שמפעילה את האפליקציה.
- `BookControllerTest.java` מחלקה שמכילה מבחנים למחלקת `BookController` כדי לוודא את נכונות הפעולות.
- `Request.java` מחלקה שמייצגת בקשה שמגיעה מהלקוח לשרת.
- `Response.java` מחלקה שמייצגת את התגובה שהשרת מחזיר ללקוח לאחר עיבוד הבקשה.

+

•

0

# פעולת המערכת - ממשק משתמש

- המערכת שלנו מאפשרת ביצוע פעולות שונות על ספרים ומשתמשים באמצעות ממשק המשתמש. LibrarianUI.java
- הוספת ספר: המשתמש מזין את פרטי הספר ומוסיף אותו למערכת. AddBookUI
- הסרת ספר מהרשימה: באמצעות removeBook
- חיפוש ספר: searchBook
- עדכון פרטי ספר: ניתן לעדכן את פרטי הספר הקיים במערכת. UpdateBookUI
- ניהול משתמשים: ניתן לנהל משתמשים (הוספת משתמש, הסרת משתמש, עדכון פרטי משתמש, חיפוש משתמש) באמצעות ממשק המשתמש. CustomerUI.java