

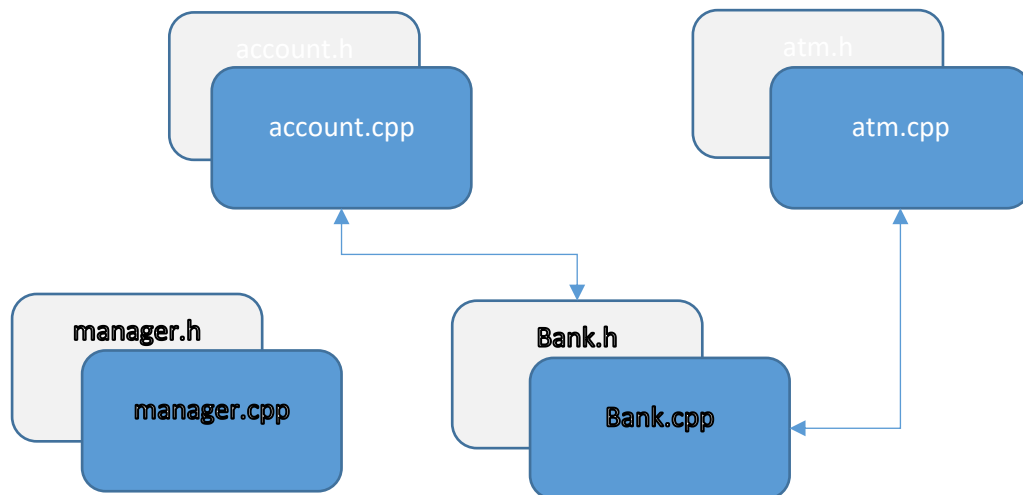
מבנה מערכות הפעלה

רטוב 2

שגיא גרש 204095313

יהב וינוקור 307853010

להלן דיאגרמה אשר מתארת את רכיבי התוכנית ומעבר הנתונים ביניהם:



תיאור החלקים העיקריים בתוכנית:

- Manager.cpp
 - מכיל את פונקציית ה-main אשר מאתחלת את התוכנית:
 - מאתחל בנק.
 - מאתחל מערך כספומטים לפי הקלט (כמות וקבצי קלט).
 - מאתחל את החוטים: חוט לכל כספומט, חוט ל-commission וחוט ל-print all.
 - מחכה לסיום חוטי הכספומטים, ולאחר מכן משחרר זיכרון ומסיים את התוכנית.
- Atm class
 - מייצג כספומט אשר בעזרתו מבוצעות כלל הפעולות.
 - הכספומט רק קורא לביצוע הפונקציה הנדרשת (על פי קובץ הקלט שלו) על ידי הבנק.
 - בעת קריאה לפעולה מהבנק, מעביר את הפרטים הנרשמים כגון סיסמה, מספר משתמש וכמות כסף למשיכה.
 - מקבל חזרה מהבנק את תוצאת הפעולה ומדפיס אותה לקובץ ה-log.
 - מקבילות: מבצע נעילה להדפסה לקובץ ה-log המשותף בלבד, לא מבצע נעילות נתונים כלל.
- Bank class
 - מכיל את מאגר של חשבונות.
 - מקבל דרישות לביצוע פעולות על חשבונות מן הכספומטים וקורא לביצוע הפעולה בחשבון הספציפי, מקבל את תוצאת הפעולה מן החשבון ומעביר את התוצאה לכספומט.
 - מדפיס את מצב הבנק.
 - מחייב עמלה רנדומלית מכל החשבונות ומכניס אותה ליתרת הבנק.
 - מקבילות:
 - מאגר החשבונות: מנגון כותבים/קוראים, יצירת משתמש הינה כתיבה למאגר, וכל שאר פעולות הבנק הינן קריאה.
 - יתרת הבנק: מנגון כותבים/קוראים, לקיחת עמלה הינה כתיבה והדפסת מצב הבנק הינה קריאה.
- account class
 - מכיל את הפרטי החשבון ומבצע פעולות עליו בהתאם לבקשת הבנק.
 - מקבל את הנתונים הדרושים לביצוע הפעולה, מוודא את נכונותם, מבצע את הפעולה ומעביר את תוצאת הפעולה לבנק.
 - מקבילות:
 - יתרת החשבון: מנגון כותבים/קוראים, בירור היתרה הינו קריאה, וכל שאר הפעולות הינן כתיבה.

מבני הנתונים בהם השתמשנו

מימשנו את התוכנית באמצעות תכנות מונחה עצמים. מימשנו שלושה קלאסים –

- Class ATM
- Class Bank
- Class Account

בחרנו לממש את התוכנית באופן הזה כי כל אחד מה classes מאופיין על ידי פעולות ומשתנים משלו, בנוסף מבין ה classes שניים צריכים להיות משוכפלים מספר פעמים – Account ו ATM.

ה Bank class מכיל מבנה נתונים של מפה של מצביעים ל class account - כאשר האינדקסים הם מספרי המשתמשים, וזה גם כדי לאפשר הדפסה ממוינת לפי מספר משתמש.

ב main, מאותחל מערך של מצביעים ל ATM class .

בנוסף אנו מחזיקים מערך של threads אשר מוקצה לכל אחד מפעולות הריצה של ה ATM, לפעולת העמלה של הבנק ולפעולת ההדפסה של הבנק, אשר נרחיב עליהן בהמשך.

פירוט הפעולות המתמשות בכל קלאס והמנשקים ביניהם:

• Account:

הקלאס מחזיק את מספר החשבון, הסיסמא והיתרה בחשבון.

קלאס המשתמש מממש את הפעולות, המתרחשות על הכסף –

Deposit, withdraw, get_balance.

בנוסף ממומשת פעולת commission על הכסף בחשבון.

הבנק מחזיק את "מפת החשבונות" ולכן יש לו גישה לפונקציות האלה, וכך נעשות על ידו פעולות על הכסף.

בנוסף ממומשות פונקציות עזר:

Get_password, על מנת לאפשר פעולות המתרחשות ללא סיסמא (למשל העברה בין חשבונות)

- כל פעולת המתרחשת על הכסף בחשבון היא פעולה קריטית ותחומה ב mutex, כאשר פעולת get_balance ממומשת כ read ושאר הפעולות ממומשות כ write.

• Bank:

קלאס המממש פונקציות אשר בעיקרן קוראות לפונקציות של ה account –

Deposit, withdraw, get_balance.

הוא מממש גם פונקציות create_account ו transfer.

הוא מקבל דרישה מה ATM ומבצע את הפעולות על החשבונות בהתאם.

בנוסף הבנק ממש את הפעולות commission ו print all אשר להן מוקצה חוט ב main, מאחר והן מתבצעות באופן בלתי תלוי לשאר ריצת התוכנית ובמקביל אליה.

שתי פונקציות עזר (*_helper) מסוג void* ממומשות ב class לצורך זה.

- ATM:

קלאס הממשק עם המשתמש.

הקלאס מממש את פונקציות הממשק להן יכול לקרוא המשתמש:

create_account | Deposit, withdraw, get_balance, transfer.

ה-ATM מחזיק מצביע לבנק, וכך קורא לפעולות על החשבונות של הבנק מתוך הפונקציות שלו.

ה-ATMs צריכים לעבוד במקביל ולכן מוקצה לכל אחד מהם חוט בחוץ main, בנוסף ממומשת פונקציית עזר בקלאס על מנת לבצע שורת קלט ב-ATM – זהו בעצם ממשק המשתמש.

בנוסף ה-ATM אחראי על ההדפסות ל-log – כפי שמשתמע מממשק עם המשתמש, ההדפסות עטופות ב-mutex על מנת שלא ישתלבו אחת בשנייה.