

תיאור Flow:

המערכת בנויה כך שקיימת תקשורת בין הלקוח לשרת ע"י מחלקות: Client-I TCPServer.

TCPServer – מחלקת שרת בה יש הרצה ויצירת שרת תוך הפעלת מתודת Run עם העברת פרמטר של MatrixHandler. מתודת Run מקבל טיפוס מסוג IHandler ויוצרת Thread נפרד שבתוכו ישנם שני Treadpools. אחד ליצירת קשרים עם לקוחות ספציפיים. השני לטיפול המשימות תוך העברת טיפוס ClientTaskServer עם אותו IHandler.

ClientTaskServer – מחלקה שמממשת Runnable. שתפקידה לטפל במשימות הלקוח. במחלקה זו יש מתודה Run בה יש הגדרת Runnable שנקרא ClientHandling מטפל במשימה ובסיומה סוגר את כל ערוצי התקשורת והמשאבים.

Client – מחלקה שתפקידה לצור socket תפעולי מול השרת, מחזיקה Input/Output Stream, מעבירה מידע לשרת ומקבלת מידע מהשרת. ישנו תפריט לניהול בקשות לקוח באמצעות SwitchCase המבדיל בין משימה למשימה.

Connected – במחלקה זו עשינו שימוש באלגוריתם DFS ע"מ למצוא את כל רכיבי הקשירות עבור משימה 1. המחלקה תקבל כפרמטר מטריצה. במחלקה זו הגדרנו מיון רכיבי קשירות עפ"י גודל ומימוש Comparator.

ShortPaths – מחלקה זו מוצאת את כל המסלולים הקצרים ביותר מאינדקס מקור לאינדקס יעד במטריצה בינארית. מחלקה מקבלת כפרמטר מטריצה, ואינדקסים מקור ויעד. במחלקה זו עשינו שימוש באלגוריתם BFS וחישוב המסלול המינימלי ומציאת כל המסלולים הקצרים ביותר.

SubMarine – ראשית נבדוק את כמות רכיבי הקשירות במטריצה, ועבור כל רכיב קשירות נמצא שטח מקסימלי. לאחר מכן בודקים האם השטח מתאים לשטח מלבן, ואם כן זוהי צוללת תקינה. כל רכיב קשירות הכולל יותר מקודקוד אחד – מועמד להיות צוללת. הבדיקה נעשית בנתיב ביצוע נפרד. יש מנגנון נעילה בקטע הקריטי כאשר יש כתיבה למשתנים. המחלקה תחזיר כי הערך לא תקין - אם כמות רכיבי הקשירות גדולה מכמות הצוללות. אם הקלט תקין – המחלקה תחזיר את כמות הצוללות.

BellmanFord – במחלקה זו עשינו שימוש באלגוריתם בלמן-פורד ע"מ למצוא מסלולים קלים ביותר עבור משימה 4. המחלקה מקבלת כפרמטר: מטריצה משוקללת, אינדקס מקור ואינדקס יעד.