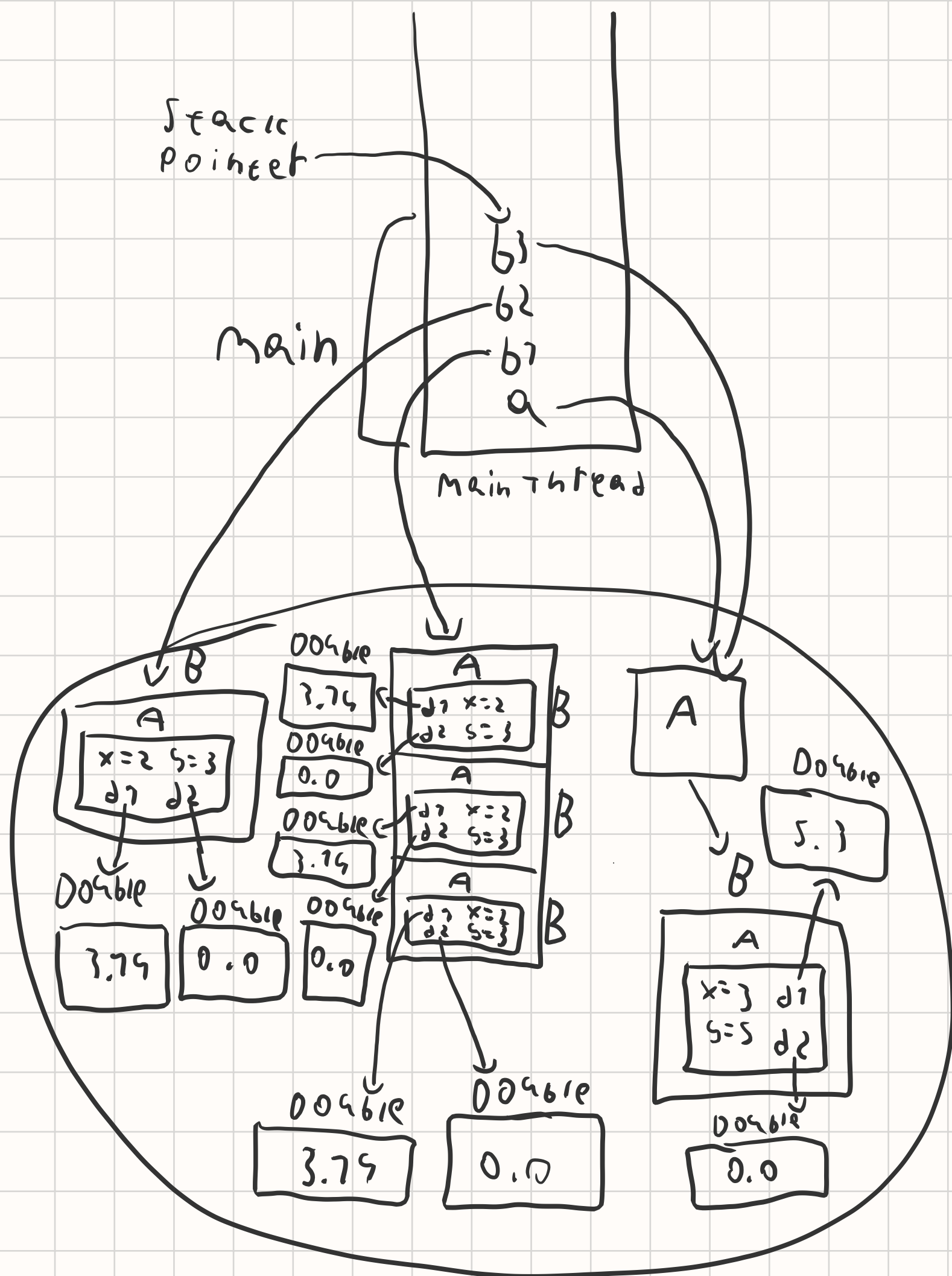


(272472484) תוצון פרמי (375924376) / ~2 11 : פר'ען
https://github.com/subsidy2032/Work4 : Github

11775 - 7 ת'ס'ע



שאלה 2 – Semaphore

הקוד הבא מדגים שימוש בסמפור. בקטע קוד זה נרצה לאפשר לשניים מתוך שלושה תהליכים לרוץ במקביל, בעוד שהשלישי יצטרך להמתין.

על מנת שהקוד יבצע את פעולתו כהלכה, יש להשלים בכל מקום שבו מופיעה הערה על השלמה נדרשת.

```
import java.util.concurrent.*;

public class SemaphoreExample {
    static class SharedResource {
        private final Semaphore semaphore;

        public SharedResource(int permits) {
            semaphore = new Semaphore(permits, true);
        }

        public void accessResource() {
            try {
                semaphore.acquire() // Acquire a permit

                System.out.println(Thread.currentThread().getName() + " is accessing the resource.");

                // Simulating some work
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            } finally {
                semaphore.release() // Release the permit
            }
        }
    }
}
```

```

static class Worker extends Thread {

    private final SharedResource sharedResource;

    public Worker(String name, SharedResource sharedResource) {
        super(name);
        this.sharedResource = sharedResource;
    }

    @Override
    public void run() {
        while(true) {
            sharedResource.accessResource();
        }
    }
}

public static void main(String[] args) {

    SharedResource sharedResource = new SharedResource(2); // permits available

    // Create multiple threads trying to access the resource
    Thread thread1 = new Worker("Thread 1", sharedResource);
    Thread thread2 = new Worker("Thread 2", sharedResource);
    Thread thread3 = new Worker("Thread 3", sharedResource);

    thread1.start();
    thread2.start();
    thread3.start();
}
}

```

JVM נשאלה 3 -

יש לתאר בקצרה את תהליך ה – Verification שבתוך שלב ה – linking.

בתהליך ה – Verification שבשלב ה – linking הJVM מוודא שהקבצי מחלקה בנויים בצורה נכונה, מאחר שיכול להיות שנעשה שימוש בקומפייילר לא מתאים או שיתעוררו בעיות לאחר קומפילציה, לדוגמה הJVM בודק כי חתימות השיטות נכונות וכי ישנה הגדרה נכונה של משתנים, במידה ויש שגיאה תיזרק חריגה מסוג `VerifyError`.