אוניברסיטת בן גוריון

הפקולטה למדעי המחשב

מבוא לפירוק והרכבה של תכניות – פרויקט סיום

*מטרת ונושא הפרוייקט: הדגמת אלגוריתמי code-motion וrefactoring שנלמדו במהלך הסמסטר.*

מרצה הקורס: ד"ר רן אטינגר

מגישים: טל יצחק (204260533)

מאי נגלר (200935963)

כלל הקוד בפרויקט נכתב בJava, וזמין ב-GitHub:

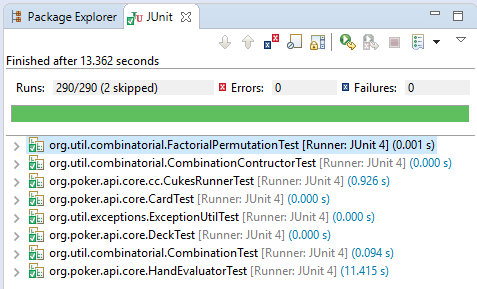
github.com/talitz/Introduction-to-Code-Motion-Refactoring-Project

סמסטר ב', תשע"ז, 2017

רקע

הפרויקט מממש Parser גמיש ומהיר ל-JSON, הזמין בשפות Java ו-Go. הפרויקט נלקח מהrepository הזמין לצפייה בקישור הבא: <https://github.com/json-iterator/java>.

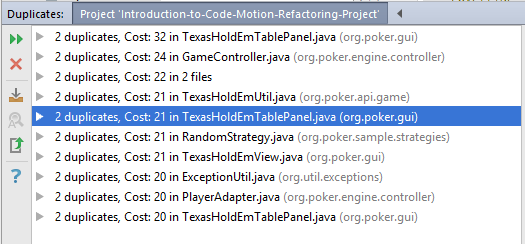
הפרויקט נכתב עם סביבת טסטים בJUnit, להלן המחשה של מעבר כלל הטסטים:



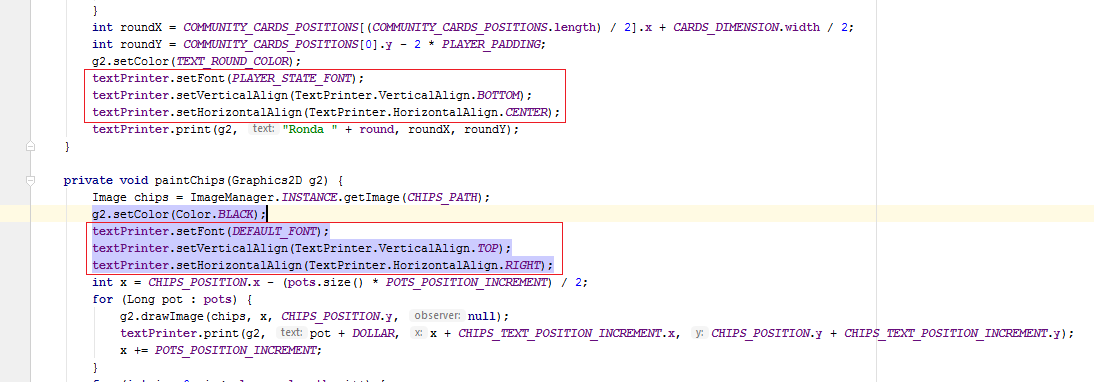
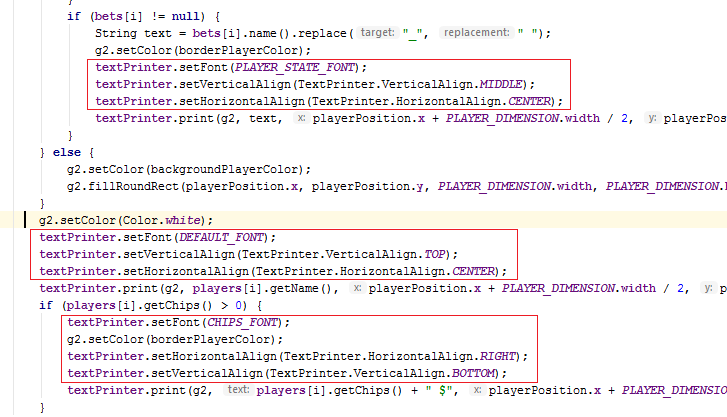
בפרויקט נציג דוגמאות מקטעי קוד נבחרים ונריץ את האלגוריתמים שלמדנו בכיתה לrefactoring & code motion.

דוגמא 1: Extract Method & PDG-Based Sliding

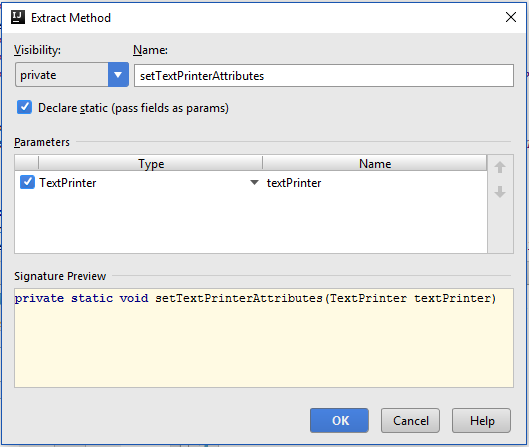
ראשית, השתמשנו בכלי של IntelliJ למציאת שכפולי קוד בתכנית, ע"י לחיצת קליק ימני על הפרויקט, Analyze -> Locate Duplicates, קיבלנו את התוצאות הבאות, מתוכן התעניינו בשכפול המסומן:



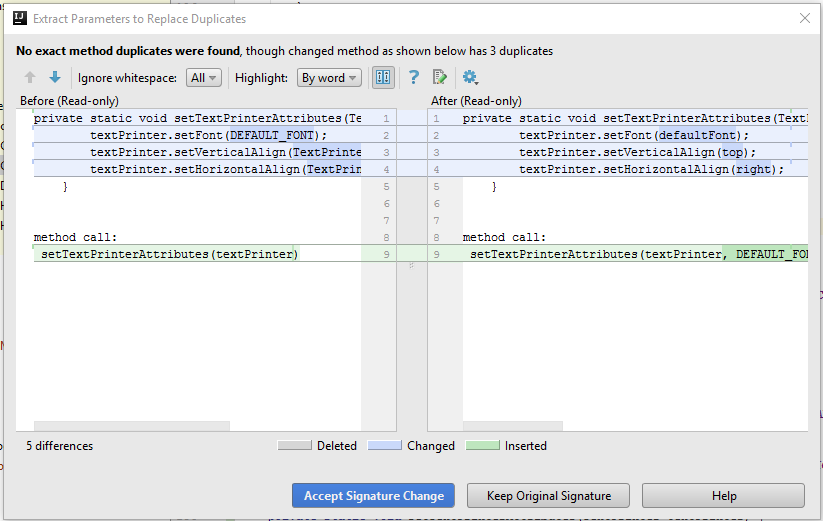
נסמן את הקוד המשוכפל בריבועים אדומים, זהו שכפול מסוג Clone Type 1.



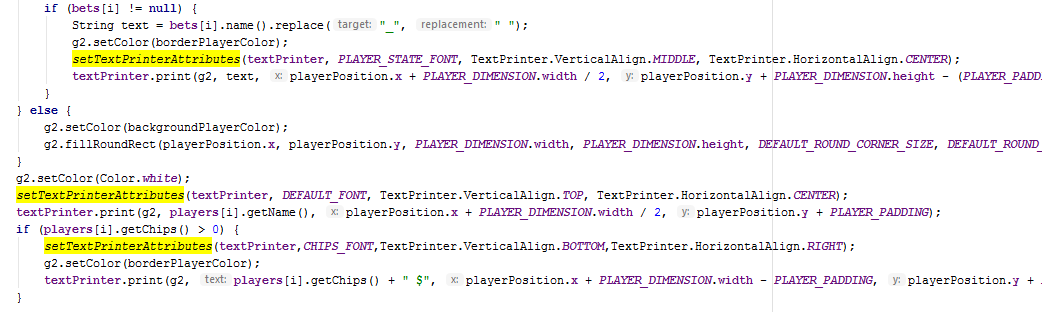
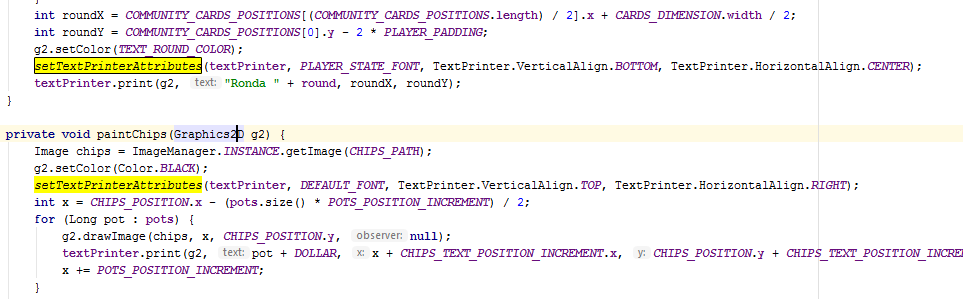
קיימות 5 חתיכות קוד שכאלה, לכן נרצה לבצע Extract Method כדי למנוע חזרות מיותרות:



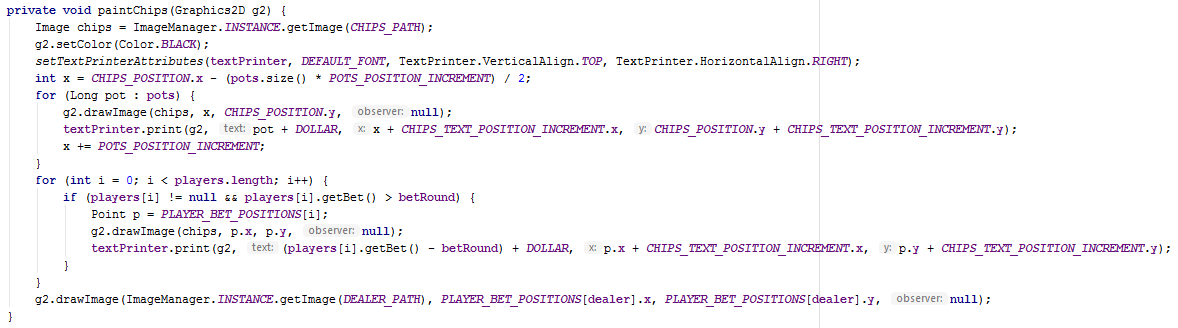
כעת, נלחץ על Accept Signature Change (שמעבירה לפונקציה את הערכים שמבצעים עליהם set):



ולהלן התוצאות:

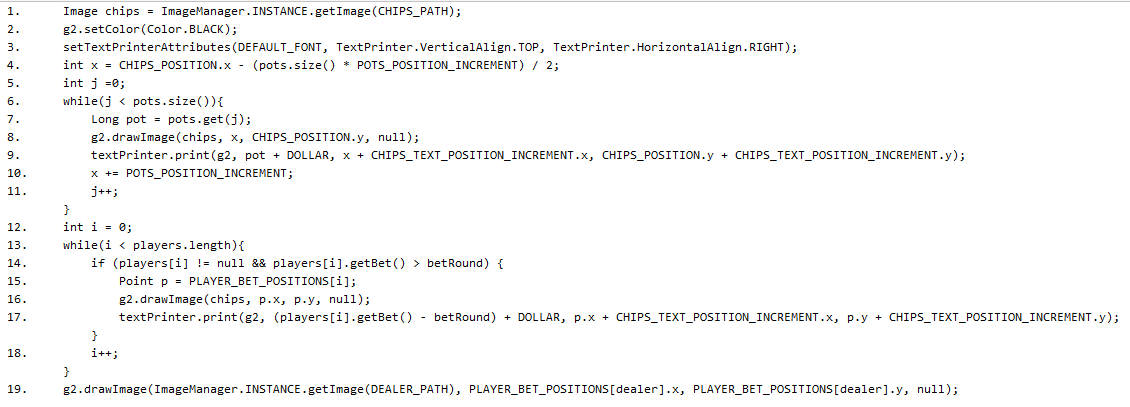


כעת, נסתכל בחתיכת הקוד הבאה:

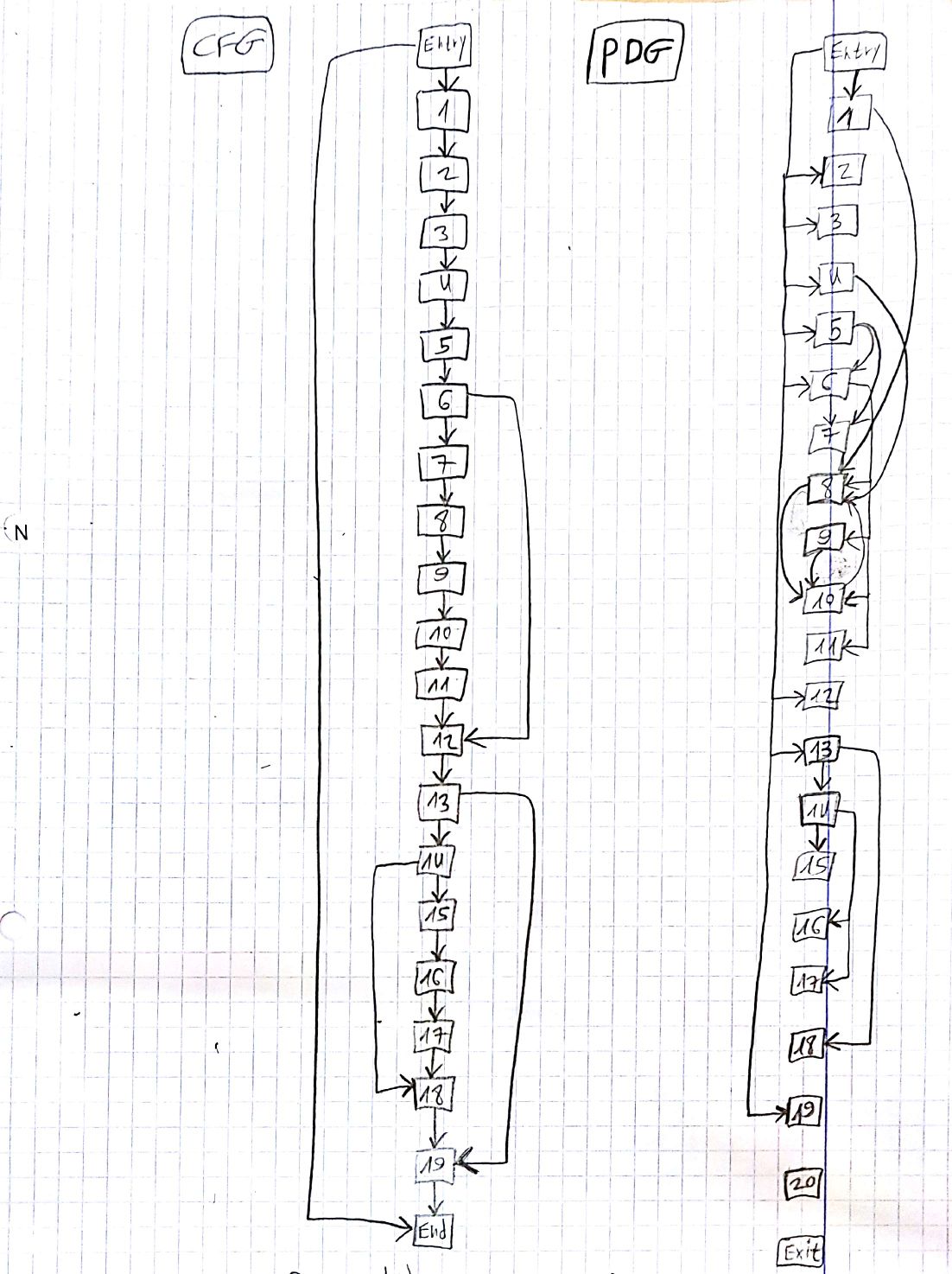


נרצה לבצע PDG-Base Sliding כאשר קבוצת המשתנים שנבחר תהיה .

נבצע המרה שקולה של לולאות הfor ללולאות while כפי שנאמר בהרצאה, כמו כן נמספר את כלל השורות בקוד:



נבנה את גרפי הCFG ו-PDG:



כבסיס לסליידינג על קטע הקוד המורכב מהפקודות 1-19 שלעיל, נקבל את הייצוג הבא: קבוצת הקודקודים N היא:

וקבוצת הקשתות E הכוללת תלויות מסוג control (הקבוצה Ec) ותלויות data (הקבוצה Ed) מסוג flow או anti כדלהלן:

|  |  |  |
| --- | --- | --- |
| Vars | Kind | Edge |
| - | Control | (Entry,1) |
| - | Control | (Entry,2) |
| - | Control | (Entry,3) |
| - | Control | (Entry,4) |
| - | Control | (Entry,5) |
| - | Control | (Entry,6) |
| - | Control | (Entry,12) |
| - | Control | (Entry,13) |
| - | Control | (Entry,19) |
| - | Control | (6,7) |
| - | Control | (6,8) |
| - | Control | (6,9) |
| - | Control | (6,10) |
| - | Control | (6,11) |
| - | Control | (13,14) |
| - | Control | (13,18) |
| - | Control | (14,15) |
| - | Control | (14,16) |
| - | Control | (14,17) |
| {chips} | Flow | (1,8) |
| {g2} | Flow | (2,8) |
| {g2} | Flow | (2,9) |
| {g2} | Flow | (2,16) |
| {g2} | Flow | (2,17) |
| {g2} | Flow | (2,19) |
| {textPrinter} | Flow | (3,9) |
| {textPrinter} | Flow | (3,17) |
| {x} | Flow | (4,8) |
| {x} | Flow | (4,9) |
| {x} | Flow | (4,10) |
| {j} | Flow | (5,6) |
| {j} | Flow | (5,7) |
| {j} | Flow | (5,11) |
| {pot} | Flow | (7,9) |
| {x} | Anti | (8,10) |
| {x} | Anti | (9,10) |
| {pot} | Anti | (9,7) |
| {x} | Flow | (10,8) |
| {j} | Flow | (11,6) |
| {j} | Flow | (11,7) |
| {i} | Flow | (12,13) |
| {i} | Flow | (12,14) |
| {i} | Flow | (12,15) |
| {i} | Flow | (12,17) |
| {i} | Flow | (12,18) |
| {p} | Flow | (15,16) |
| {p} | Flow | (15,17) |
| {i} | Flow | (18,13) |
| {i} | Flow | (18,14) |
| {i} | Flow | (18,15) |
| {i} | Flow | (18,17) |
| {textPrinter} | Flow | (3,Exit) |
| {textPrinter} | Flow | (9,Exit) |
| {textPrinter} | Flow | (17,Exit) |
| {g2} | Flow | (2,Exit) |
| {g2} | Flow | (8,Exit) |
| {g2} | Flow | (16,Exit) |
| {g2} | Flow | (19,Exit) |

כעת נפעיל סליידינג עבור .

תוצאת הסליידינג:

1. הקוד המחולץ (הסלייס, תת הקבוצה Nv) יהיה:
2. הקוד המשלים את הקו-סלייס יהיה (Ncov):

נבצע Extract Method לslice ול-co-slice, ונקבל:



נוודא כעת שכלל הטסטים עדיין עוברים לאחר השינוי:

