

עבודת הגשה 2

דגשים להגשה

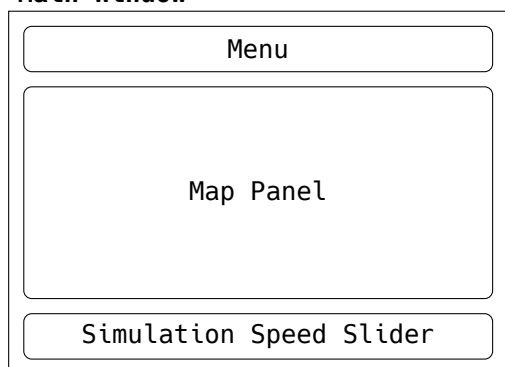
- ניתן להגיש את העבודה בזוגות או בבודדים, בהתאם להגשה של המטלה הראשונה.
- בתיעוד של כל קובץ java חובה לציין את השמות ומספר תעודת הזהות של המגישים.
- חובה לתעד כל קובץ, כל מחלקה וכל מתודה על ידי תיעוד javaDoc.
- יש להגיש את העבודה עד **2021-05-02** דרך Moodle בלבד, בתור קובץ zip בודד. במקרים של הריון, מילואים, אשפוז יש לספק אישור מתאים על כך, עד 48 שעות לפני מועד ההגשה.
- לשאלות, ניתן לפנות במייל כשבפנייה חובה לציין שם פרטי ושם משפחה והסבר מפורט של השאלה. הפניות יבוצעו אל המתרגל ארתור זמירין, במייל arthuza@ac.sce.ac.il

דגשים למטלה זו

- זוהי מטלה לימודית בלבד ואינה מנסה להתקרב לייצוג המציאות. אין לקחת מסקנות על אופי המחלה על בסיס סימולציה זאת! הנוסחאות נבחרו לפי תאימות לדרישות שלנו ולא לפי נתונים אמיתיים.
- יש לבנות היררכיית מחלקות בצורה הטובה ביותר, ולהימנע משכפול קוד.
- המטלה הנוכחית מתבססת על המטלה הקודמת ומהווה בסיס למטלות הבאות, ולכן השקיעו בתכנון ובביצוע כדי למנוע שכתוב מלא.
- ניתן להוסיף מחלקות עזר, ואף ניתן להוסיף שדות ומתודות למחלקות לפי צורך. חשבו היטב כיצד ומה להוסיף והיזהרו משבירת כימוס ונתינת גישה לא נדרשת.
- על כל השדות בכל המחלקות להיות פרטיים בלבד. צריך לבנות Setters רק אם קיומם נחוץ. Settres שרק לתתי-מחלקות אמורה להיות גישה אליהן, יהיו protected.
- ודאו שבכל אתחול כל השדות מאותחלים בערכים, כלומר שאין אפשרות לקבל מופע במצב לא חוקי.
- לכל מחלקה מתאימה יש לממש בנוסף מתודת toString ומתודת equals.

1 ממשק גרפי – חבילת UI

Main Window



1.1 חלון ראשי

החלון הראשי מורכב ממספר חלקים, בהתאם לתמונה משמאל:

- (א) תפריט אפשרויות שונות – יתואר בהמשך.
- (ב) פאנל לציור המפה בסימולציה – יתואר בהמשך.
- (ג) [JSlider](#) המאפשר לבחור את מהירות הסימולציה.

1.2 תפריט עליון בחלון הראשי

התפריט יהיה מטיפוס [JMenuBar](#), ובו יהיו התפריטים הבאים:

(א) File:

1. Load – מציג חלון בחירת קובץ, טוען מפה חדשה מתוכו (מתקשר למחלקה `SimulationFile`). מאפשר רק אם הסימולציה לא התחילה או נעצרה (בשאר המקרים אי אפשר לבצע טעינה).
2. Statistics – פותח את אותו חלון הסטטיסטיקה (לא Modal), מאפשר רק אם בוצעה טעינה.
3. Edit Mutations – פותח את חלון עריכת מוטציות (כן Modal).
4. Exit – סוגר את כל הממשקים, עוצר את הסימולציה וכו'.

(ב) Simulation:

1. Play – משהה את הסימולציה. מאפשר רק אם הסימולציה מושהה ברגע.
2. Pause – משהה את הסימולציה. מאפשר רק אם הסימולציה רצה ברגע.
3. Stop – עוצר ומפסיק לחלוטין את הסימולציה, ומונע המשך ריצה עתידית עד הטעינה הבאה. מאפשר רק לאחר טעינת קובץ וכל עוד לא עצרנו.
4. Set Ticks Per Day – פותח Dialog (כן Modal) פשוט שקולט מהמשתמש מספר שלם חיובי ומציב אותו במשתנה הסטטי החדש במחלקת `Clock`. עבור הקלט יש להשתמש ב-[JSpinner](#).

(ג) Help:

1. Help – פותח Dialog (כן Modal) המציג מידע עזר שמסביר על שימוש בתוכנה, מציג רקע וכו'. מטרתכם להציג בחלון הזה מידע נוסף שיעזור להשתמש בתוכנה.
2. About – פותח Dialog (לא Modal) המציג מידע על שמות כותבי המטלה, תאריך כתיבתה וכו'.

1.3 פאנל ציור המפה

פאנל עליו יש לצייר את כל היישובים והחיבורים ביניהם. כל יישוב מצויר בתור מלבן, כאשר המיקום נקבע בהתאם למיקום היישוב (לפי השדה `location` ביישוב), וכאשר הצבע הוא לפי צבע הרמזור של היישוב. כל חיבור בין יישובים מצויר בתור קו ישר שחור המחבר את אמצעי מלבן היישוב. על מנת שהציור יהיה יפה, יש לצייר קודם כל את כל החיבורים ורק אחרי זה את המלבנים. בתוך המלבן של כל יישוב צריך לצייר מחרוזת שהיא השם של היישוב. בעת לחיצה על המפה, יש למצוא האם הלחיצה בוצעה מעל יישוב, ואם כן, עבור יישוב זה יש לפתוח את חלון הסטטיסטיקה כשהשורה עבור היישוב נבחרה.

1.4 חלון הסטטיסטיקה

Statistics Window

תת-חלון שתפקידו להציג חיתוך מצב נוכחי מלא של כל היישובים, המורכב ממספר חלקים, בהתאם לתמונה משמאל:

א) JComboBox המאפשר לבחור לפי איזו עמודה לסנן, כאשר תוכן הרשימה הן שמות העמודות. יש להציג רק עמודות שהגיוני לסנן על פיהן (למשל אין טעם לסנן לפי מספר אנשים חולים).

ב) תיבת טקסט שבה המשתמש יקליד את מילות הסינון שלו. הסינון יתבצע לפי מסנן מבוסס ביטוי רגולרי. אם הטקסט המוזן בתיבה אינו ביטוי רגולרי תקין, אין לשנות את הסינון הנוכחי.

ג) החלק העיקרי של החלון – טבלת הנתונים. הטבלה מאפשרת מיון, סינון, תצוגה יפה יותר ושמות משמעותיים לעמודות. כל שורה בטבלה מייצגת יישוב בודד. הטבלה יכולה להציג כל נתון שאתם רואים לנכון, אך לפחות צריכה להציג שם יישוב, טיפוס יישוב (עיר/מושב/קיבוץ), צבע רמזור, אחוז בטוח [0 ... 1] של אנשים חולים מכלל האוכלוסייה בעיר, מספר מנות חיסון שניתנו ביישוב, מספר נפטרים ביישוב, מספר האנשים ביישוב כרגע.

ד) כפתור שמירה – בעת לחיצה עליו מציג חלון לבחירת קובץ שמירה (אפשר להשתמש ב-FileDialog או ב-JFileChooser), ולאחר בחירה של המשתמש למיקום, ישמור בו קובץ סטטיסטיקה נוכחי, בעזרת המחלקה StatisticsFile.

ה) כפתור הוספת חולים – עבור היישוב הנבחר בטבלה (לפי השורה שנבחרה) גורם ל-0.1% מהאוכלוסייה הבריאה ביישוב לחלות בווריאנט אקראי.

ו) כפתור שליחת חיסונים – פותח חלון שבו המשתמש ירשום מספר חיובי של מנות חיסון, וליישוב הנבחר בטבלה תתווסף לשדה כמות מנות חיסון זאת.

1.5 חלון עריכת מוטציות

חלון שמורכב כולו מטבלה בודדת, כאשר כותרות העמודות והשורות הן שמות הווריאנטים השונים שיש בסימולציה, ובכל תא יש CheckBox המאפשר מוטציות של הווריאנטים. חובה להשתמש בחלון הזה ב-JTable. בשביל להגדיר כותרות לשורות, יש להשתמש במחלקת עזר המצורפת בסוף המטלה RowedTableScroll. שורה מייצגת ווריאנט נוכחי, וכל עמודה לאיזה ווריאנט הוא יכול להתפתח.

2 תוספות למחלקות

2.1 חבילת Country, מחלקת RamzorColor

הוסיפו לכל צבע של רמזור שדה מטיפוס Color המייצג את הצבע שאתו נצייר את המלבן על המפה. יש גם להוסיף שדה להסתברות מעבר ביישוב: ליישוב ירוק 100%, צהוב 80%, כתום 60%, אדום 40%.

2.2 חבילת Country, מחלקת Settlement

יש להוסיף שדה המייצג מספר אנשים מרבי שיכולים להיות ביישוב, מספר מנות חיסון ביישוב (מתחיל מאפס), ומערך מטיפוס Settlement המייצג את כל היישובים המחוברים ליישוב הנוכחי. את מערך האנשים (אפשר להשתמש ב-ArrayList) יש לשמור בעזרת רשימת אנשים חולים ורשימת אנשים לא חולים (יש לוודא שבעת מעבר אדם הוא מוכנס למערך הנכון מבין שניהם).

בעת ניסיון מעבר של אדם מיישוב ליישוב, כישלון יכול לקרות בגלל:

א) הגעה לקיבולת מרבית ביישוב (מספר אנשים כרגע ביישוב).

ב) לפי צבע הרמזור, יש לקחת את מכפלת הסתברויות המעבר ביישוב הנוכחי שלו והיישוב מטרה, ולפי מכפלה זו להחליט האם המעבר הצליח.

2.3 חבילת IO, מחלקת SimulationFile

קיבולת מרבית של יישוב נקבעת בתור מספר האנשים המוגדרים בהתחלה כפול הקבוע 1.3. בקובץ הנטען, המגדיר את מפת הסימולציה, פרט לתיאור הקובץ שהופיע במטלה הקודמת, קיים טיפוס חדש של שורות, המתחילות בסולמית. שורה כזאת מגדירה חיבור בין שני יישובים. הפורמט של השורה הוא 3 חלקים מופרדים בנקודה פסיק: סולמית, שם יישוב אחד ושם יישוב שני. עבור כל חיבור כזה, צריך להגדיר את החיבור בכל אחד משני היישובים (להוסיף את האחר למערך החדש שהוספתם). שימו לב שאין הכרח ששורת חיבור יישובים תופיע אחרי שהוגדרו היישובים עצמם, ולכן מומלץ שקודם כל תאתחלו את היישובים ואת החיבורים תשמרו, ורק אחר כך תגדירו את החיבורים ביישובים.

2.4 חבילת IO, מחלקת StatisticsFile

מחלקה המאפשרת לשמור את חיתוך מצב המחלה בכלל היישובים, כאשר פורמט הקובץ הוא csv, כלומר טבלת ערכים, כאשר השורות בטבלה מופרדות בירידת שורה, והעמודות בטבלה מופרדות בפסיקים. עליכם לשמור בקובץ את כל הנתונים המופיעים בטבלת הסטטיסטיקה, לפי המיון והסינון שהופיעו ברגע הלחיצה על כפתור השמירה. כדי לבדוק את הפלט, פתחו את הקובץ ב-Excel.

2.5 חבילת Simulation, מחלקת Clock

המשתנה הסטטי במחלקה, המייצג את הזמן, נמדד ב-tick-ים. הוסיפו משתנה שלם סטטי הנקרא ticks_per_day, המייצג את מספר ה-tick-ים ביום אחד. המשתנה הזה יכול להשתנות מהתפריט (הוגדר מקודם), וכברירת מחדל ערך המשתנה הוא אחד. הוסיפו פונקציה המקבלת זמן התחלה, ומחשבת כמה ימים עברו מזמן ההתחלה עד עכשיו (בזמן החילוק יש לעגל למעלה). בכל נוסחאות החישוב התלויות ב-"ימים" השתמשו בפונקציה זאת.

2.6 חבילת Virus

כאשר מתבצעת הדבקה מאדם חולה בווריאנט כלשהו, במקום להדביק בדיוק באותו ווריאנט כמו במטלה הקודמת, הפעם הווריאנט יכול להתפתח כמוטציה לווריאנט אחר. ההתפתחויות נקבעות לפי חלון עריכת המוטציות. נכון למטלה הנוכחית, בצעו בצורה הכי טובה שאתם מצליחים (כנראה שלא תצליחו לבצע את זה לפי עקרונות תכנות מונחה עצמים, הצפייה שזה יעבוד כרגע ובהמשך תלמדו איך צריך). בחישוב האם אדם יכול להדביק, אם עברו פחות מ-5 ימים מרגע ההדבקה של האדם החולה, יש להחזיר הסתברות אפס (הדבקה יכולה לקרות רק אחרי 5 ימים).

3 הגדרת הסימולציה החדשה

ההגדרה הבאה, המתארת את עיקר ריצת המערכת, אינה באה לחייב אתכם להגדיר הכל במחלקת Main בודדת. מותר ואף חובה לפרוס את כל הגדרות וחיבור ריצת הסימולציה בין המחלקות השונות על מנת לשמור על עקרונות הכימוס, הסידור והקריאות.

המילה tick משמעה הרצת סימולציה בודדת על כל היישובים. עליכם לבצע sleep לאחר כל tick (זמן השינה נקבע לפי ה-Slider בחלון הראשי), כאשר בסוף כל tick יש לקדם את ה-Clock באחד. בכל tick יש לבצע בדיקה האם אנחנו רצים כרגע (טעונים ולא מושהים), לעבור על כל אחד מהיישובים, ולבצע:

- בכל יישוב תדגמו 20% חולים אקראיים, ולכל אחד דגמו 3 אנשים לא חולים, ונסו לבצע הדבקה.
- בכל יישוב יש להפוך את כל החולים, שעברו 25 ימים מרגע ההדבקה, למחלימים.
- בכל יישוב יש לדגום 3% (חולים ולא חולים) שהם מנסים לעבור ליישוב שכן אקראי, ואז יש לנסות לעבור ליישוב הנבחר.
- אם ביישוב יש מנות חיסון ממתונות ואנשים מטיפוס Healthy, יש לחסן אותם (וכמובן לעדכן את כמות המנות). אין צורך להתעסק בעניין שתי מנות, כל אחד מקבל מנה אחת.

א' נספח: מחלקת RowedTableScroll

```
public class RowedTableScroll extends JScrollPane {
    private static class RowHeaderRenderer extends JLabel
                                                implements ListCellRenderer<String> {
        RowHeaderRenderer(JTable table) {
            setOpaque(true);
            setBorder(UIManager.getBorder("TableHeader.cellBorder"));
            setHorizontalAlignment(CENTER);
            JTableHeader header = table.getTableHeader();
            setForeground(header.getForeground());
            setBackground(header.getBackground());
            setFont(header.getFont());
        }

        @Override
        public Component getListCellRendererComponent(JList<? extends String> list,
            String value, int index, boolean isSelected, boolean cellHasFocus) {
            setText((value == null) ? "" : value.toString());
            return this;
        }
    }

    public RowedTableScroll(final JTable table, final String[] rowHeaders) {
        super(table);

        final JList<String> rowHeader = new JList<String>(new AbstractListModel<String>() {
            public int getSize() {
                return Math.min(rowHeaders.length, table.getRowCount());
            }

            public String getElementAt(int index) {
                return rowHeaders[index];
            }
        });
        rowHeader.setFixedCellWidth(50);
        rowHeader.setFixedCellHeight(table.getRowHeight());
        rowHeader.setCellRenderer(new RowHeaderRenderer(table));
        this.setRowHeaderView(rowHeader);
    }
}
```

ובעת על מנת להשתמש במחלקה הזאת, במקום שורת הקוד `this.add(new JScrollPane(table))` המופיעה בקוד המוצג במצגת, יש לבצע `this.add(new RowedTableScroll(table, col_array))` כאשר `col_array` הוא מערך מחרוזות המייצג את כותרות השורות (במקום 0 במערך היא הכותרת של השורה הכי ראשונה). גודל המערך צריך להיות לפחות כמספר השורות הצפויות להיות (במקרה שלנו במטלה הוא ידוע מראש).