

תוכנית הכשרות - צוות תוכנה

9 בספטמבר 2020

הערות

1. תחת ההנחה שכל הכשרה היא הערך שעתיים ולמצטרפים אין ידע קודם.
 2. התכנון הוא גם ליצור מצגת לכל הכשרה, בהתבסס על הנקודות שאראה כאן.
 3. שוב, קשה מאוד לחזות את המהירות שבה יילמדו את זה. מה שמתואר כאן הוא כנראה די אופטימיסטי.
- חשוב לציין שזוהי תוכנית הכשרות שנועדה להעמיק כמה שיותר: פשוט כי אני מרגיש כי זאת הדרך הכי טובה ליצור אנשים שהדברים האלה באים להם בטבעיות, ושמוסוגלים לזום ולעשות יותר עם תוכנה מסתם קוד פשוט לרובוט. למרות זאת, במקרה ויש מחסור בזמן, ניתן להנמיך את כמות הזמן באופן משמעותי (מהי שפת תכנות, סוגים פרימיטיביים)
4. המטרה היא גם לעשות סימולציה, וויז'ן ובינה מלאכותית השנה, שכרגע לא אכלול כאן, מפאת העובדה שאין לי דרך לבנות להן הכשרות. זה די בעייתי ולכן כדאי להתחיל לעבוד על הדברים האלה - כדאי שנדע את רמת המורכבות ונדע אותם באופן כללי - כמה שיותר מהר. כנראה כדאי לבקש מגרינבליץ את המצגת שלהם על סימולציות גם לפני שהם ממש עושים את ההכשרה, פשוט כי כרגע הם לא אמרו מתי הם עושים אותה, וגם באופן כללי לדבר עם קבוצות אחרות ולראות אם משהו ממש יכול לבוא ולהעביר לנו הכשרה על זה - בלי להיות מחויבים ללו"ז של גרינבליץ. כרגע נראה לי שהדרך הכי טובה לעשות את זה היא פשוט שיהיו הכשרות אופציונליות יותר בזמנים שונים. יש גם לציין שהאידיאל הוא תת-צוותים, ולכן יש סיכוי טוב שבכלל נרצה שזו תהיה התוכנית רק לחלק מהאנשים, וחל אחר יילמדו, נניח, פיית'ון ועוד יילמדו סימולציות (כתלות בכמות המצטרפים שנקבל). השורה התחתונה היא שכדאי להשלים את הפערים בידע שלנו לפחות עד ספטמבר.

Part I

Java

1 מושגי בסיס

1.1 הכשרה #1

1.1.1 מהי שפת תכנות? שפת מכונה (בינארית) מול שפת סף (Assembly Language) ושפה עילית¹. [50-30 דק']

1. מכונת טיורינג - פירוש וכתיבה של מערכת בינארית (אין/יש).

¹הבנה טובה של מה בעצם קורה פיזית (וגם תוכניתית) מהקוד שנכתב היא די חשובה להבנה שלמה וטובה של תכנות ובהחלט של תכנות רובוטים. למרות זאת, יש לציין שאני פחות מכיר את הנושא. אם משהו מאלקטרוניקה / משהו מקבוצה אחרת יוכל לבוא לעזור עם זה, יהיה מגניב.

2. איך המחשב שלנו עובד? מה הוא "מבין"?

(א) אין/יש זרם חשמלי - זה חשוב במיוחד לרובוטיקה באופן ספציפי: כמעט כל המידע שנקבל על המצב וסביבתו של הרובוט יהיה ביש/אין זרם וכמה הוא חזק.

(ב) תאי זיכרון...

(ג) ניסיון לתכנת בבינארית ישירות כנראה יהיה מאוד בעייתי - יהיה קשה מאוד לזכור את כל המחרוזות של 0 ו 1 - ובטח שלשים לב לטעויות. לכן המחרוזות הבינאריות מוחלפות בפקודות באנגלית פשוטה - שפת סף (או Assembly). למרות שהיא קלה לכתובה הרבה, הרבה יותר מבינארית שפת סף מתייחסת, עדיין, רק לשינוי של החומרה עצמה (העברה של זרם חשמלי לתאי זיכרון).

(ד) לכן, היא בדרך כלל דורשת ידע רב של חומרה והרבה מאוד שורות של קוד בשביל פעולות בסיסיות. כדי לקצר ולהקל על תהליך הכתיבה, קיימות שפות עיליות - השפות בהן, אגב, רובן רובן המוחלט של כל התכנות נכתב: הן לא מתייחסות בכלל* לחומרה הממשית, ומתרכזות במרחב דיגיטלי ומופשט יותר: מספרים, מילים, ואמיתות (input > 16 וכדו').

(ה) גם בתוך שפות העיליות יש מושג של גובה: שפות נמוכות יותר קרובות יותר לשפת מכונה, ובעצם "מכסות" פחות קוד בעצמן. שפות גבוהות יותר כחוקות יותר משפת מכונה ו"מכסות" יותר קוד בעצמן - תוכנות ומסגרות עבור כל מיני פיצ'רים באים כבר עם השפה.

1.1.2 IDEs - Integrated Development Enviornments [דק' 3/15]

1. כאמור, טעות הקלדה אחת מונעת מכל הקוד לעבוד. עכשיו, דמיינו שאתם צריכים למצוא את שגיאת ההקלדה בוורד בתוך 1000 מילים שונות או שהוא לא קריא. לא כיף במיוחד. או יעיל. לכן, אנחנו משתמשים במשהו שנקרא סביבת פיתוח משולבת - או, בשם פחות מסורבל, IDEs - *Integrated Development Enviornment*.
2. אם אין מספיק מחשבים - > התקנה של VS Code והרחבות של Java < [15 דקות]. (אם יש², מצגת עם הוראות התקנה תישלח אחרי השיעור. [3 דק'])

- יש לציין שעדיין לא נתקין את ההרחבות של פירסט, כי בשלב זה אין צורך.

1.1.3 הדפסה למסך: `system.out.println` [10-20 דק']

1. יצירת פרויקט VS Code Hello World

(א) הסברת מבנה הקובץ:

i. כל תוכנה מוכלת בתוך משהו שנקרא מחלקה בשם זהה לשם הקובץ, אשר נפתחת ב `public class` `<name>` כשאחריהם סוגריים מסולסלות. נלמד מה כל המילים האלה אומרות בשיעורים מאוחרים יותר.

ii. בכל תוכנה ישנה פונקציה אחת ראשית הנקראת `main` - שם כל הקוד שממש רץ כתוב. היא נפתחת ב"`public static void main(String[] args)`", כששוב, אחריהם בוגריים מסולסלות. גם את משמעות המילים האלה נלמד בשיעורים מאוחרים יותר.

iii. כאמור, בתוך `main` נמצא הקוד האמיתי (סוף-סוף): הפונקציה `system.out.println`, אשר מציגה (להלן, "מדפיסה") על המסך את המילים ששמים בסוגריים שלנו.

- הקוד שלנו, בעצם, מדפיס למסך את המילים "Hello World!" - מסורתית זה הקוד הראשון שכותבים בשלומדים לתכנת.

2. תרגיל: פתחו קובץ ריק וכתבו קוד שמדפיס ברכת שלום מכס.

- הדגש הוא על העבודה ב VS Code - פתיחת תיקייה, קובץ שנגמר ב `.java`, וקיצורי `main`, `sysout`.

3. יצירת פרויקט של ממש בניגוד לקובץ Java יחיד - הפרויקט גם מזהה שגיאות.

²לפחות להכשרה הראשונה, כדאי מאוד שנהיה בחדר מחשבים / אדריכלות.

1.1.4 קלט - `system.console().readline()` וסוגי קונסולות [12-17 דק']

1. התאמה למשתמש - אנחנו כנראה רוצים קלט. נשתמש ב`system.console().readLine()`.

- `System.out.println("Hello, " + System.console().readLine("Name: "));`
- שימו לב לפלוס: ניתן לצרף מילים (`concatenate`).

2. נראה שהתוכנה נכשלת: `NullPointerException` - ניסינו לפנות לערך שלא קיים.

- זה קורה בגלל שבאופן אוטומטי הקוד משתמש בקונסולת הדיבאגינג, שמיועדת לתת לנו מידע על מה שקורה בקוד ולא לאפליקציות של ממש. כדי לתקן את השגיאה, נוסיף לקובץ `launch.json`:
"console": launch.json
"integratedTerminal" - שמשנה את הקונסולה שהתוכנה משתמשת בה לטרמינל, בה ניתן להזין מידע ופקודות, כמו `cmd` בווינדוס.

3. בכל מקרה, עכשיו הקוד עובד. אבל הוא נראה די מסובף ומכוער ובא לי שהוא יהיה בשתי שורות -