

תוכנית הכשרות - צוות תוכנה

9 בספטמבר 2020

הערות

1. במקום להכין כאן תוכנית הכשרות מסודרת לגמרי (תוך אידיעה של כמות זמן וקצב למידה), במקום זה הכנתי "שיעורים" - שהם פשוט כל מיני תתי-נושאים שכדאי ללמד ביחד. הם לא יותר משעה במקסימום ויותר קרובים לחצי שעה, אז יש הרבה גמישות. כן תכננתי את ההכשרות עצמן, פחות או יותר - מבחינת הסדר ואיזה שיעורים יהיו בכל אחד, שזה כל מה שנשאר, אני חושב - ושיבצתי אותם כאן [קישור]. אפשר לשנות את השיעורים לכל הכשרה ושאר העמודות יתעדכנו אוטומטית. לשיקולכם.
2. התכנון הוא גם ליצור מצגת לכל שיעור. הן ייכתבו לפני כל הכשרה עם ספייירים, פשוט כי זה לוקח המון זמן לעשות אותם ביחד לפני.
3. שוב, קשה מאוד לחזות את המהירות שבה יילמדו את זה.
 - עבור כל שיעור, כתבתי טווח זמן משוער, בעיקר ע"פ האינטואיציה שלי. אני לא באמת יכול להבטיח קורלציה למציאות, במיוחד שהתרגולים לא בנויים כ"כ עדיין אז קשה להעריך כמה זמן הם ייקחו.
 - שיעורי הבית העיקריים שהם יקבלו זה לעבור על דברים באפליקציה ללימוד תוכנה (sololearn). אם הם יחליטו ללמוד מנה כשממעמם להם (ויכול להיות מאוד שזה יקרה), זה יכול לשנות מאוד את הקצב שלנו. למרות שגם בדברים שהם ילמדו שם אני רוצה להמיק, זה יאיץ מאוד את ההבנה ויוריד ממני את הצורך ללמד הרבה דברים ואת הסינטקס שלהם - זה יקצר את הזמן שצריך באופן די דרסטי. אני מניח שנראה.
 - 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, ושמוסגלים לזום ולעשות יותר עם תוכנה מסתם קוד פשוט לרובוט. למרות זאת, במקרה ויש מחסור בזמן, ניתן להנמיך את כמות הזמן באופן משמעותי (אפשר לקצר את ההכשרה הראשונה לאפילו פחות משעה - אבל זה עולה בעומק ורוחב, ואלא אם כן יש סיבה של ממש, נראה לי שלא כדאי לשלם את זה).
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 הכשרה #1 (חומרה, מושגים בסיסיים)

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

1. מכונת טיורינג - פירוש וכתיבה של מערכת בינארית (אין/יש).
2. איך המחשב שלנו עובד? מה הוא "מבין"?
3. איך הקוד שאנחנו כותבים מיתרגם לחשמל?

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

1. מהם IDEs?
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.

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

1. יצירת פרויקט Hello World בVS Code והסברת מבנה הקובץ.
 - הסברת הפקודה `system.out.println`
2. תרגול: פתיחת קובץ ריק וכתיבת קוד שמדפיס ברכת שלום.
 - הדגש הוא על העבודה בVS Code - פתיחת תיקייה, קובץ שנגמר בjava, וקיצורי `main`, `class`, `sysout`.
3. יצירת פרויקט של ממש בניגוד לקובץ Java יחיד - הפרויקט גם מזהה שגיאות.

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

1. פקודת `system.console().readLine()`.
2. הרכבת ממחרזות עם פלוס.
3. `NullPointerException` - למה קיבלנו אותה: קונסולת דיבאגינג מול טרמינל.
 - תיקון ב`launch.json`
4. העברת מידע בין שורות הקוד דרך משתנה `String`.

1.5 סוגי נתונים [5-7 דק']

1. מספר, מילה ואמיתה, וההבדלים ביניהם.
-
- 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 מישהו מאלקטרוניקה / מישהו מקבוצה אחרת יוכל לבוא לעזור עם זה, יהיה מגניב.
- ²לפחות להכשרה הראשונה, כדאי מאוד שנהיה בחדר מחשבים / אדריכלות.

1.6 מספר חלק 1 - מספר ממשי [10-15 דק']:

1. מתי נרצה להשתמש במספר?
2. Double ו Float - הבדלים ושימוש.
3. המרת מחרוזת לממשי.
4. אופרטורים בסיסיים: חיבור, חיסור, כפל וחילוק, + תרגול קצרצר
5. אופרטורים מתקדמים: ספריית Math ופונקציות min, max, pow, וטריגו.
6. מילת final.

1.7 בדיקת שוויון - switch case [25-40 דק']

1. מתי נשתמש בזה?
2. סינטקס.
3. תרגול - מחשבון

1.8 מספר חלק 2 - שלמים [20-40]:

1. מתי משתמשים: ספירה מול ספקטרום.
2. byte, short, int, long, ההבדלים ביניהם ומתי משתמשים בכל אחד.
3. המרת מחרוזת לשלם.
4. פונקציית רצפה (floor division) ומודולו (שארית): מה זה, איך משתמשים בזה ומתי משתמשים בזה.
5. המרת שלם לממשי ולהפך.
6. אופרטור ++
7. תרגול

2 הכשרה #2 (מושגים בסיסיים, סדרות ולולאות 1)

2.1 מילים: תו ומחרוזת [7-12 דק']

1. תו: כיצד הוא מאוחסן, המרה לASCII ולהפך, פונקציות
2. מחרוזת: מהי, איך יוצרים אותה. חיבור מחרוזות.

2.2 אמיתות (booleans) [30-50 דק']

1. מהו בוליאן? מה הוא מייצג? אילו ערכים הוא יכול להכיל?
2. מה השימוש? אופרטורים בסיסיים (=, <, >), פקודות if, else, תרגול קצר (גילאים)
3. אופרטורים מתקדמים (&&, (and), ||, (or)
4. פקודת else if.
5. תרגול (אלרגיות, טווחים).

2.3 לולאות [30-50 דק']

1. לולאות while, do while, while אינסופית, תרגול (do while - משתמש צריך לתת מספר חיובי. לשאר אפשר לקחת דוגמאות מהספר).
2. לולאת for + תרגול (סכום, ממוצע, סדרה, עוד מהספר).

3 הכשרה #3 (סדרות ולולאות 2)

3.1 מערכים [30-50 דק']

1. מהם מערכים? מתי נשתמש? דוגמא + סינטקס בסיסי (type[] name = {value, value.. value}).
2. סינטקס מתקדם - הכרזה ללא ערכים, השמת ערכים לפי מיקום, השבת ערכים לפי מיקום, קבלת מיקום לפי ערך. תרגול קצרצר תוך כדי.
3. איטרציה - for וforeach (נכלל גם תכונת length), סיכום (דוגמא).
4. מחרוזות כמערכים.
5. תרגול.
6. מערכים דו-מימדיים: מתי נשתמש (מיני-מחלקות), סינטקס.

4 הכשרה #4 (תכנות מונחה עצמים 1, אלגוריתמיקה)

4.1 מתודות ואלגוריתמיקה³ [2-3 ש']

1. מהי אלגוריתמיקה? למה אנחנו צריכים אותה? אבסטרקציה של פעולות לחזרות של פעולות בסיס. ככל שנכיר יותר את השפה ונתרגל יותר, כך יהיה לנו יותר קל לעשות אלגוריתמיזציה מתאימה. עצה לאלגוריתמיזציה: לנסח במילים הכי פשוטות שאפשר.
2. עם דוגמאות והסברים ממשחק הזיכרון [קישור] (לכל אחד: סינטקס, כיצד הם עוזרים באלגוריתמיזציה):
 - (א) מתודות void ללא פרמטרים
 - (ב) מתודות void עם פרמטרים
 - להסביר על שינוי פרמטרים בפונקציה: פרמטרים מועברים כערכים ולא כמצביעים (יש דרך אבל נדבר עליה בפעם אחרת)
 - (ג) החזרה: מתודות מחזירות פרימיטיבים / מחרוזות.
3. פרמטרי ברירת מחדל
4. העמסת יתר בדוגמת דרייבטריין: מהירות שווה ומהירות לכל מנוע.
5. עוד על אלגוריתמיזציה
- החשיבות בלהבין מה הפונקציות שבהן אנחנו משתמשים עושות.
6. תרגול:

- (א) לכתוב פונקציית and שלוקחת מערך ומחזירה בוליאן לפי אופרטור && (בלי להשתמש בו)
- (ב) לכתוב פונקציית or שלוקחת מערך ומחזירה בוליאן לפי אופרטור || (בלי להשתמש בו)
- (ג) לכתוב פונקציית הכפלה שלוקחת שני מספרים ממשיים ומחזירה את המכפלה הממשית שלהם (בלי להשתמש ב*)
- (ד) לכתוב פונקציית חילוק שלוקחת שני מספרים ממשיים ומחזירה את המנה שלהם (בלי להשתמש ב /)
- (ה) לכתוב פונקציית מודולו שלוקחת מספר ממשי ומספר שלם ומחזירה את השארית (בלי להשתמש ב %)

³יש לציין שהשיעור לא ספציפי לג'אבה, וחשוב מאוד לתכנות ככלל. בנוסף, אעדיף מאוד שלמרות שהוא מאוד ארוך, הוא כן ילומד בהכשרה אחת. אם לא, אעדיף לפחות לא ללמד דברים אחרים באותה הכשרה.

7. ריקרציה

(א) הסבר וסינטקס

(ב) באלגוריתמיקה: נוסחה ריקרסיבית מול נוסחה מפורשת (לפי $f(n-1)$ מול לפי n , דוגמאות + חזקות של שתיים, עצרת.

(ג) חסרונות: קשה להבין, הרבה פעמים פחות איטי. לדוגמא,

8. תרגול אלגוריתמיקה נוסף כללי

5 הכשרה #5 (תכנות מונחה עצמים 2)

5.1 מבוא לתכנות מונחה עצמים [1:15-1:45 ש']

1. מהו תכנות מונחה עצמים? למה צריך אותו?

- קשר לרובוטיקה.

- עצמים כסט של תכונות

2. סינטקס בסיסי (בלוק `class`, `import` ויצירת אובייקט, אופרטור נקודה), עצמים כסוג נתונים.

3. תרגול קצר (יצירת מחלקות בסיסיות לשלושה אובייקטים בחדר)

4. מתודות עם אובייקטים:

- מתודות של מחלקות - הסבר, תרגול קצר - הוספה לקוד מהתרגול הקודם

- אובייקטים כפרמטרים: הסבר, סינטקס.

– אובייקטים מועברים כמצביע ולא כערך. העברה של פרימיטיבים כמצביעים באצעות מחלקות, דוגמא, תרגול קצר.

5.2 תכנות מונחה עצמים מתקדם [1:00-1:30 ש']

1. אובייקט סופי (final) - שינוי תכונות מול השמה.

2. הרשאות גישה: `private` ו `public`: למה צריך? (סדר, הגנה), חוסר מילת הרשאה (רק בחבילה).

3. מתודות `set` ו `get` -

- לבדיקה או שינוי של קלט / פלט

- ליישום יותר קל של שינויים (גם אם אין צורך בהגנה, בדיקה או שינוי)

4. פעולות בונות (constructors)

- קונסטרוקטורים ללא פרמטרים

- קונסטרוקטורים עם פרמטרים: השמת תכונות.

- העמסת קונסטרוקטורים.

- תרגול.

5. משתנים סופיים ומחלקות / משתנים / שיטות סטטיות + תרגול

6. מחלקות `enum` - הסבר, סינטקס, דוגמא מהקוד של 2019

6 הכשרה #6 - תכנות מונחה עצמים 3

6.1 תורשה

1. מה זה תורשה? למה צריך את זה? סינטקס והסבר בסיסי (נוח יותר)