

## הנחיות לכתיבת ממי"נים

---

במסמך זה מרוכזות הנחיות לכתיבת הממי"נים בקורס "מבוא למדעי המחשב ושפת ג'אווה". אנא קראו את המסמך בעיון. שימו לב שחלק מההערות מתייחסות לחומר מתקדם בקורס, ולכן הן יהיו רלוונטיות לכשנגיע ליחידות הרלוונטיות.

### 1. תיעוד

תיעוד הוא מרכיב חשוב בכתיבת קוד. התיעוד מאפשר לכם להסביר את עצמכם בשפה אנושית – למה התכוונתם, איך מימשתם את הקוד, מה האלגוריתם וכו'. התיעוד מקל על קורא חיצוני להבין את הקוד, ומקל גם עליכם לקרוא את הקוד שלכם אחרי זמן. מספר כללים לכתיבת תיעוד –

1. בראש כל מחלקה יש לכתוב תיעוד שמפרט את שם המחלקה, שם הכותב, תאריך הכתיבה וכן תיאור קצר על מהות המחלקה (מה היא מייצגת, מה היא עושה וכו').
  2. לפני כל שיטה יש לתעד את הפרמטרים שהשיטה מקבלת ומה תפקידם, מה תפקיד השיטה ומה הפלט שלה.
  3. לפני כל קטע קוד מורכב יש לכתוב ולהסביר מה עושה קטע הקוד הזה ואיך.
  4. כל התיעוד נעשה באנגלית בלבד.
- שימו לב – אין צורך להרבות במלל סתם, רק כדי להגיד שיש תיעוד. התיעוד, כאמור, הוא כלי חשוב לקריאות התוכנית ולא אמור להכביד על הקורא. לכן אין טעם לתעד שורות ברורות מאליהן כמו למשל שורות הדפסה, קלט פשוט או דברים בסגנון זה.
- החל ממטלה 13 תצטרכו להוסיף גם תיעוד מיוחד של מחלקות שנקרא API (java doc). ביחידה 7 פרופ' רוזנשיין מסביר על צורת תיעוד זו.

### 2. שמות

בשפת תכנות אנחנו משתמשים בהרבה שמות שאנחנו ממציאים לחלקים מהקוד – שמות של מחלקות, משתנים, תכונות ושיטות. כיוון שבתוכנית נורמלית ישנם הרבה שמות, רצוי וראוי שהשמות יהיו משמעותיים ככל הניתן ויביעו את מה שהשם מתאר.

אנא הקפידו לבחור שמות לפי מוסכמות הכתיבה בג'אווה הנהוגות בקורס זה, וכן להעדיף שמות משמעותיים. **השמות חייבים להיות באנגלית ממש ולא בעברית באותיות אנגליות.**

**קובץ ובו המוסכמות של כתיבת שמות בקורס זה נמצא בספרית הקבצים להורדה באתר, בתת-ספריה "מדריכי עזר".**

### 3. קבועים (finals)

קבועים הם משתנים שמקבלים ערך רק פעם אחת במהלך חייהם, ואינם יכולים להשתנות במהלך התוכנית. אנו משתמשים בקבועים לשמירת ערכים מספריים שחוזרים בקוד, כאשר שמירת הקבוע מקלה על קריאת הקוד, הבנתו ותחזוקתו. למשל, אם אנו כותבים תוכנית שמבצעת חישובים על ציוני סטודנטים בקורס מסוים, ואמורה להוציא כפלט כמה סטודנטים עברו וכמה נכשלו בקורס, כדאי להגדיר קבוע בשם, נניח, FAIL\_GRADE שיציין את הציון המינימלי הדרוש למעבר קורס, ולא להשתמש במספר 60 כמו שהוא בתוך הקוד.

### 4. עימוד וריווח

שפת ג'אווה היא שפה ש"עיוורת לרווחים", כלומר, בדר"כ היא לא מחייבת כתיבת רווחים, שורות חדשות, טאבים או כל סימן אחר. מצד שני, אנחנו בני האדם מאוד לא עיוורים לרווחים, וקשה לנו לקרוא תוכנית שכתובה בצורה לא מסודרת. לפיכך אנו נוהגים לפי מוסכמות העימוד הבאות –

1. הזחה (אינדנטציה indentation) – כל בלוק חדש של קוד ייכתב בהסטה של טאב (Tab) ימינה. למשל, אחרי חתימת השיטה, לוחצים על טאב, וכותבים את הקוד שם. בסוף השיטה חוזרים אחורה. אחרי כתיבת משפט if לוחצים על טאב וכותבים את הקוד שם, ובסוף התנאי חוזרים אחורה. לדוגמא, אחת משתי הדרכים להלן מקובלת:

<pre>public int something (int x) {     x = x+1;     return x; }</pre>	או	<pre>public int something (int x) {     x = x+1;     return x; }</pre>
--	----	--

2. שורות חדשות – כל פקודה שנגמרת בנקודה פסיק תיכתב בשורה משלה. אנא המנעו מלכתוב מספר פקודות באותה שורה, זה לא קריא.

אחרי כתיבת משפט if לוחצים על טאב וכותבים את הקוד שם, ובסוף התנאי חוזרים אחורה. לדוגמא, לא לכתוב כך:

```
if (x>0) x++;
```

אלא כך: (והפקודה שתהיה אחרי משפט if תחזור לעמודה מתחת לאות i)

```
if (x>0)
    x++;
```

כל כותרת (חתימה של שיטה, הצהרת משתנים, הצהרת מחלקה) תכתב בשורה משלה. לפני כל בלוק קוד חדש כדאי לשים שורת רווח.

3. רווחים – הקפידו לרווח בין מילים, אופרטורים לוגיים וחשבוניים, סוגריים וכו'.

## 5. שכפול קוד

בתכנה אנחנו משתדלים לא לכתוב את אותו הדבר פעמיים, ולא לכתוב דברים שכבר כתבנו ושכבר עובדים. אם כבר קיימת שיטה שמבצעת את הדרוש לנו – עדיף להשתמש בה ולא "להמציא את הגלגל מחדש". אי שימוש בחלקי קוד קיימים וכתבתם מחדש נקרא **שכפול קוד** ויש להימנע מכך.

## 6. תכונות של מחלקה

תכונות של מחלקה הן משתנים שמאפיינים את המחלקה וקובעים את מצבם של העצמים. לפיכך המשתנים שנקבעים כתכונות חייבים להיות **באמת** תכונות, ולא סתם משתנים זמניים שמשמשים שיטות ספציפיות. ככלל אצבע אתם יכולים לשאול את עצמכם האם משתנה באמת מאפיין את המחלקה? אם לא, כנראה שהוא לא אמור להיות תכונה. בפרט, משתנים שמשמשים שיטה ספציפית בחישוב ספציפי ואינם מאפיינים את המחלקה לא אמורים להיות תכונות. משתנים כמו איטרטורים של לולאות, מונים זמניים וכו' אינם אמורים להיות תכונות.

## 7. עמידה בדרישות הממשק והוראות ההגשה

ברוב הממ"נים אנו נבקש שתכתבו דברים ספציפיים בשמות ספציפיים. עליכם לכתוב את שמות השיטות והמחלקות **בדיוק** כפי שנכתבו בהוראות ההגשה. אין לשנות את השמות ואת החתימות של השיטות. בנוסף, אין להוסיף שיטות public מעבר למה שהוגדר בממן. ככלל אין מניעה להוסיף שיטות פרטיות (private) כאוות נפשכם, אלא אם צוין אחרת.

אם נדרש מהתכנית שלכם להדפיס משהו, ההדפסה חייבת להתאים לפורמט הנתון התאמה שלמה, מדויקת ומלאה.

בנוסף, בסוף כל ממ"ן מצורפות הנחיות הגשה. עליכם לקרוא אותן בעיון ולפעול לפיהן – אנא הגישו **רק** את הקבצים הדרושים בממ"ן, כשהם ארוזים בקובץ zip **יחיד** ללא תוספות כל שהן. (zip ולא rar!) בדקו שהקובץ zip נפתח באופן תקין, ורק אז שלחו אותו.

## 8. קומפילציה (הידור)

החל מממ"ן 12 תתבקשו להגיש תכניות, כלומר קבצים של מחלקות. **המחלקות שתגישו חייבות להיות מקומפלות**. מחלקה שלא תעבור קומפילציה לא תיבדק ותקבל ציון 0. שימו לב שניתן לבדוק האם תכנית עוברת קומפילציה בלחיצת כפתור אחת, ועל כן כאמור לא תיבדק תכנית שאינה מתקמפלת. כמו כן, באחריותכם לבדוק שהתכנית אכן עובדת ועושה את מה שהיא אמורה לעשות. אתם צריכים לבדוק גם מקרי קצה ולבדוק שלא יהיו שגיאות ריצה.

אגב, כדאי להגיש חלק מתכנית כאשר הוא כתוב נכון וטוב, במקום לנסות לענות על הכל ולגרום לתכנית כולה לא להתקמפל, (ואז ממילא התרגיל אינו בר־בדיקה).

שימו לב שעליכם להגיש את הקבצים שהסיומת שלהם היא java, כלומר, קובצי הקוד של המחלקות, ולא את הקבצים המקומפלים שהם עם סיומת class. בדקו היטב את מה שאתם שולחים לפני שאתם שולחים זאת.

**ועוד משהו חשוב מאד –**

**כאשר אתם מגישים מטלה במערכת, אם עשיתם זאת בצורה נכונה, אתם מקבלים מספר אסמכתא של המטלה במערכת. שמרו לעצמכם מספר אסמכתא זה! כך אפשר לעקוב אחרי מטלה ש"נעלמת" פתאום. אם אין לכם מספר אסמכתא, סימן שתהליך ההגשה היה שגוי והמטלה לא הועלתה למערכת.**

**שימו לב, כל מה שכתבנו כאן בא לעזור לכם לכתוב נכון את התכניות בקורס שלנו בכלל, ואת המטלות בקורס בפרט. עם זאת, כל הכתוב כאן מחייב, ואנו נוריד נקודות (לפעמים אפילו הרבה) למי שלא יעמוד בסטנדרטים הכתובים כאן.**