**תנאים**

עד כה למדנו שקוד רץ בצורה כרונולוגית, מלמעלה למטה.  
כלומר, כל שורות הקוד רצות לפי הסדר בדיוק, זאת אחר זאת, כך שכל שורת קוד בתורה תתבצע ותרוץ בתורה.

לפעמים נרצה לגרום למחשב לבצע פקודות רק בהתאם **למצבים** מסוימים.

**לדוגמה:**

תכנית הקולטת מהמשתמש את גילו ובהתאם מדפיסה האם הוא יכול להצביע בבחירות או לא.

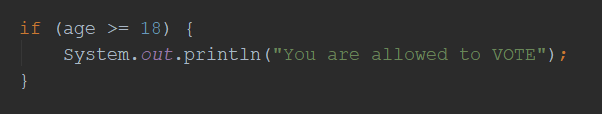
* במידה והגיל שהכניס המשתמש גדול או שווה ל 18 – ניתן להצביע בבחירות.
* אחרת, (הגיל קטן מ18) – התוכנית לא תדפיס כלום.

כשנרצה לכתוב קוד **מותנה** נשתמש במילת התנייה ממש כמו באנגלית. המילה השמורה שjava משתמשת בה היא if. נכתוב את התנאי ולאחר מכן נצהיר על פקודת ביצוע במידה והוא מתממש.

If = האם.

שימו לב כיצד ניתן לפתור את בעיית ההצבעה בבחירות המוזכרת למעלה בעזרת **תנאים**:

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&ds=true&dsyoff=20px&dsblur=68px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=import%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520static%2520public%2520void%2520main(String%2520%255B%255D%2520args)%2520%257B%250A%2520%2520%2520%2520%2520%2520System.out.println(%2522Please%2520enter%2520your%2520age%2522)%253B%250A%250A%2520%2520%2520%2520%2520%2520Scanner%2520myScanner%2520%253D%2520new%2520Scanner(System.in)%253B%250A%2520%2520%2520%2520%2520%2520int%2520age%2520%253D%2520myScanner.nextInt()%253B%250A%250A%2520%2520%2520%2520%2520%2520if%2520(age%2520%253E%253D%252018)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522You%2520are%2520allowed%2520VOTE%2522)%253B%250A%2520%2520%2520%2520%2520%2520%257D%250A%2520%2520%2520%2520%257D%250A%257D%250A)

ניתן לראות כי שלאחר קליטת הערך (גיל) מהמשתמש כמו שהכרנו, נוספה שורת קוד חדשה:

בשורה זו מתבצעת בדיקה על המשתנה, ולאחר הבדיקה נפתח 'בלוק' (=קטע קוד התחום בתוך סוגריים מסולסלים { }).

**מה יקרה בקטע קוד הזה?**

לאחר קליטת הערך מהמשתמש, המשתנה age יכיל את הערך שקיבלנו בקלט.

כשהקומפיילר יגיע לפקודת הif הוא יבדוק האם **התנאי** שבתוך הסוגריים ( ) אכן מתקיים.

**תנאי –** יכול להיות רק ערך בוליאני – true/false

לכן,

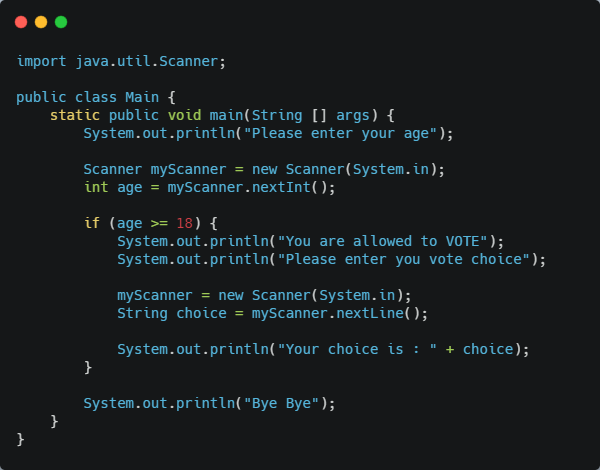
* **במידה ובמשתנה age אכן יש ערך אשר גדול או שווה ל18**  
  הקומפיילר ייכנס לתוך הבלוק ויריץ את הפקודות הכתובות בתוך הבלוג

במקרה שלנו – ידפיס את ההודעה הנתונה.

* **במידה ובמשתנה age יש ערך קטן מ18**

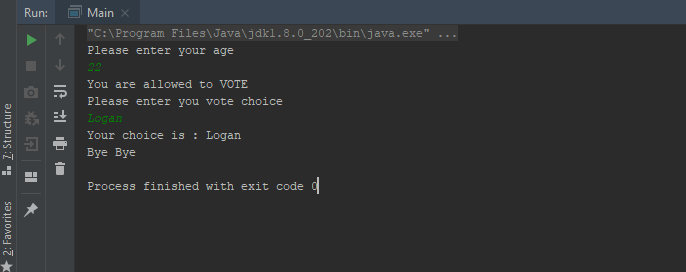
הקומפיילר לא ייכנס לתוך הבלוק, וידלג עליו.  
הקוד ימשיך לרוץ מאחרי הבלוק – ההודעה לא תודפס.

ניתן להכניס איזה קטע קוד שנרצה בתוך הבלוק, הוא ירוץ במידה והתנאי מתקיים:



כל קטע הקוד שבתוך הבלוק { } ירוץ במידה והתנאי מתקיים (במידה והגיל גדול/שווה ל18).

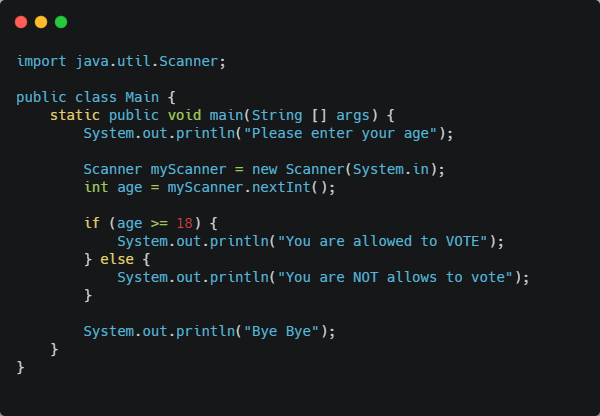
דוגמה לפלט:



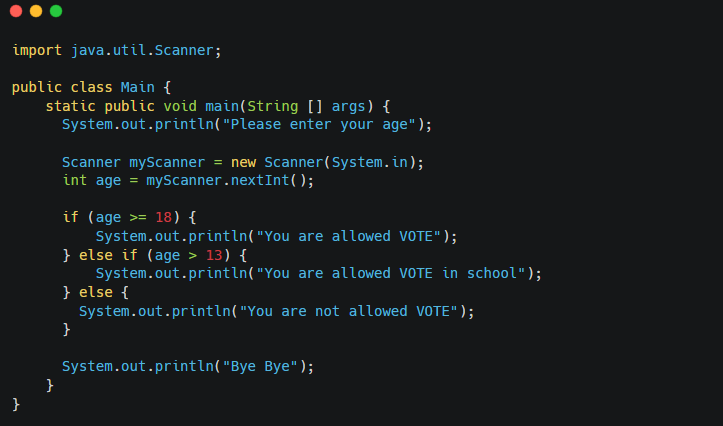
מה אם נרצה להדפיס הודעה ספציפית במידה והתנאי אינו מתקיים?  
כלומר במידה והגיל הוא פחות מ18 להדפיס הודעה שאין עוד זכות להצביע.

אם אתם לא סגורים על איך מבצעים השוואה (מה ההבדל בין = ל==) זה הזמן לחזור לשיעור אופרטורים. שימו לב שבתוך התנאי יוכל להיכתב גם תנאי מורכב ככל שיהיה באמצעות אופרטורי או ווגם

בדיוק בשביל זה יש לנו גם את פקודת **else**.  
else = אחרת



במידה והגיל **מעל 18**  
תודפסנה הפקודה שבתוך בלוק   
הif. אחרת, (דהיינו כל שאר האופציות הנותרות) תודפסנה הפקודה שבתוך בלוק הelse.

במידה והתנאי הראשון שלילי וארצה לבדוק תנאי אחר במקומו ניתן אף להשתמש בפקודת' **else if**' ולאחת אותן באופן הבא: 

ישנם שלושה מצבים אפשריים.

* במידה והגיל מעל 18, נדפיס הודעה שניתן להצביע.
* במידה והגיל בין 13 ל 18 נדפיס הודעה שניתן להצביע רק בביה"ס.
* אחרת, נדפיס הודעה שלא ניתן להצביע כלל.

כמובן שיכולו להצטרף עוד פקודות else if רבות למבנה.

לעיתים נרצה לבדוק מספר תנאים ולבצע פעולה/פעולות מסוימות בהתאם לתנאי רלוונטי בודד שמתקיים/לא מתקיים. לדוגמה ארצה לקבל מהמשתמש מספר בין 1 ל7 ולהדפיס את היום בשבוע המתאים לו. עקרונית אפשר היה לכתוב שבעה תנאים בידיעה שרק אחד מהם יהיה נכון, נשמע מסורבל קצת לא?. בדיוק לצורך כך קיימת אופציית **switchCase** בJava.

תוכלו בדוגמה המצורפת לראות את דרך השימוש בswitch.

אם ברצף תנאים כל פעם כתבנו את התנאי כולו (משתנה והשוואה לערך) הפעם:

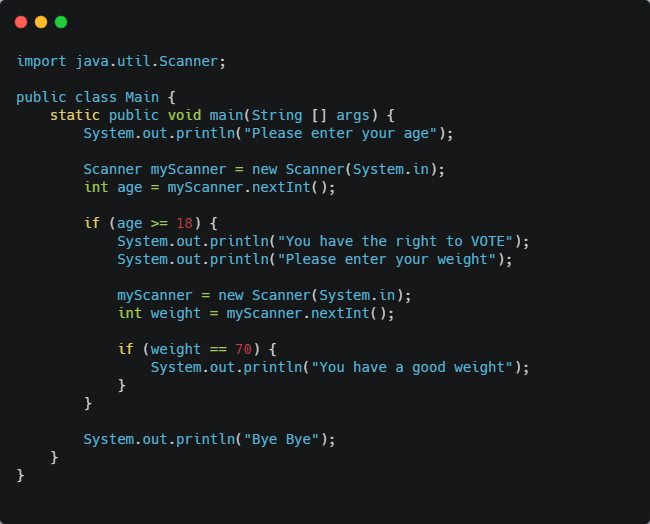
לאחר המילה השמורה switch נכתוב בסוגריים את המשתנה עליו נשאלת השאלה ואותו נשווה לכל אחד מהערכים. בכל אחד מהcaseים שלנו נציין את הערך אליו נשווה ומיד לאחר מכן את רץ הפקודות שאנחנו רוצים לבצע במידה ויש שיוויון. לאחר כל רצף פקודות נסיים בהכרח בפקודה break שאומרת שאנחנו לא ממשיכים לפקודות הבאות.

בסוף כל הcaseים נוסיף שדה default שיתפקד כמו case עם פקודות שיתרחשו במידה ואף אחד מהערכים הקודמים לא שווה.

פתרון באמצעות switch case  
הינו אלגנטי ומתאים יותר למצבים בהם יש מספר אופציות גדול הידועות מראש.

**תנאים מקוננים**

היות וניתן לכתוב איזה קוד שאנו רוצים בתוך בלוק של תנאי כלשהו, ותנאי בעצמו הוא קוד לכל דבר.  
ניתן להגדיר גם תנאים בתוך תנאים בתוך תנאים וכן הלאה.

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=import%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520static%2520public%2520void%2520main(String%2520%255B%255D%2520args)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Please%2520enter%2520your%2520age%2522)%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520Scanner%2520myScanner%2520%253D%2520new%2520Scanner(System.in)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520age%2520%253D%2520myScanner.nextInt()%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520if%2520(age%2520%253E%253D%252018)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522You%2520have%2520the%2520right%2520to%2520VOTE%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Please%2520enter%2520your%2520weight%2522)%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520myScanner%2520%253D%2520new%2520Scanner(System.in)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520int%2520weight%2520%253D%2520myScanner.nextInt()%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520if%2520(weight%2520%253D%253D%252070)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522You%2520have%2520a%2520good%2520weight%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Bye%2520Bye%2522)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)

שימו לב כי בדוגמה הנ"ל הגדרנו **תנאי בתוך תנאי.**  
בנוסף, השתמשו באופרטור ההשוואה '==' אשר לא מבצע השמה של ערכים, אלא רק משווה ביניהם.  
האופרטור '==' מחזיר true/false בהתאם.

במידה ונכנסו לתנאי הראשון, רצנו על הפקודות, הגענו לתנאי השני, ונכנסנו לתנאי השני, אז תודפס ההודעה הרלוונטית.

**תרגול**

**שימו לב בחלק מהשאלות בתרגול תצטרכו להשתמש באופרטור השארית %. האופרטור מחזיר את השארית של המספר משמאל לאחר חלוקה במספר לימין לדוגמה – התוצאה של התרגיל 43 % 5 היא 3. (מנת החלוקה היא 8 והשארית 3) 18 % 6 == 0 זה TRUE**

1. כתוב תוכנית שמקבלת מספר, מעגלת אותו לעשרת הקרובה ביותר ומדפיסה את התוצאה (ספרת יחידות חמש ומעלה תעוגל כלפי מעלה). אין להשתמש בחומר שלא נלמד עד כה. יש לפתור רק באמצעות פעולות חיבור, חיסור ושארית. אין צורך להתייחס למספרים שליליים.
2. כתוב תוכנית הקולטת שני מספרים ומדפיסה הודעה האם אחד מהם הוא גורם מחלק של השני (המספר מתחלק בו עם שארית 0 - אין חשיבות לסדר).
3. כתוב תוכנית אשר קולטת ציון במבחן מהמשתמש, ומדפיסה תיאור מילולי של הציון עפ"י הטבלה הבאה:

|  |  |
| --- | --- |
| **ציון** | **תיאור מילולי** |
| 0 - 55 | Failed |
| 56 - 69 | Almost enough |
| 70 - 79 | Passable |
| 80 - 89 | Good |
| 90 - 94 | Very good |
| 95 - ומעלה | Excellent |

לא מובטח שהקלט תקין.  
במידה והציון לא תקין יש להדפיס:  
Grade must be greater than 0.

במידה והציון גבוה מ100 **בנוסף** להדפסת המילה Excellent תודפס הודעה: קיבלת X נקודות בונוס. כשX זה ההפרש בין הציון ל100.  
שימו לב לבצע את הקוד יעיל ככל האפשר.

1. בעל מוזיאון רוצה לבדוק האם מספר קבוצות זהות בגודלן של אנשים יכולים להיכנס למוזיאון מבחינת תפוסה בכל יום. כתוב תוכנית המקבלת כמות אנשים מקסימלית במוזיאון, את כמות הקבוצות שמתוכננות לאותו יום, וכמות אנשים בקבוצה. בהתחשב בעובדה שלכל קבוצה מתווסף מאבטח ומדריך (וגם הם חלק מהתפוסה) התוכנית תדפיס האם יש מספיק מקום לכל האנשים המתוכננים להגיע לאותו יום. במידה ואין תדפיס לכמה אנשים אין מקום.
2. עלתה דרישה מהמכון המטארולוגי לתוכנית ממוחשבת שתקבל **טמפרטורה שנתית ממוצעת** במדינה מסויימת ותחשב לפי עונת השנה את **הטמפרטורה העונתית** הצפוייה. כתוב תוכנית שמקבלת טמפרטורה ממוצעת בשנה באיזור מסויים במעלות צלזיוס. לאחר מכן היא תקלוט את "קוד עונת השנה" 1 – חורף, 2 – אביב, 3 – קיץ, 4 סתיו. ותפעל על פי החישובים הבאים:
   1. בחורף – התוכנית תשאל את המשתמש האם החורף גשום ותצפה לתשובה: "1" אם כן "0" אם לא. בחורף הטמפרטורה העונתית נמוכה ב5 מעלות מהממוצעת, ואילו בחורף גשום נמוכה ב7 מהממוצעת.
   2. באביב – הטמפרטורה העונתית קטנה ב10 אחוז מהממוצעת.
   3. בקיץ – התוכנית תשאל מה אחוזי הלחות, במידה ואחוזי הלחות נמוכים מ60, הטמפרטורה העונתית גבוהה ב8 מעלות מהשנתית ממוצעת, אחרת הטמפרטורה העונתית תהיה גבוהה ב12 מעלות מהממוצעת.
   4. בסתיו – הטמפרטורה העונתית זהה לטמפרטורה הממוצעת.

בסופו של דבר התוכנית תדפיס את הטמפרטורה העונתית הצפוייה. מובטח קלט תקין.

1. שבע בום - כתוב תכנית הקולטת מספר בין 1 ל-999. הקלט לא מובטח תקין. במידה והקלט לא בטווח מדפיסה "Invalid input" .  
   במידה ותקין מדפיסה את המילה: BOOM אם המספר מתחלק ב-7 או כולל בתוכו את המספר 7. במידה ולא, תדפיס: NO BOOM

**בהצלחה!**