מטלה 2 – שאלה 9

אלעזר פיין

א. המחלקה ArrayList מממשת ממשק Iterable - ולכן מספקת אובייקט מסוג Iterator אשר נותן פונקציונליות מעבר על האיברים שנמצאים במערך המוחבא שלה. בפקודה המצוינת אנחנו פותחים instance של Iterator כזה ושומרים אותו במשתנה it. כעת ניתן לעבור על איברי הרשימה באמצעות פקודות כמו ()hasNext ,()next.

ב. הפקודה מחזירה את האובייקט הבא באיברים שאנחנו עוברים עליהם וזורקת חריגה כאשר אין עוד איברים לעבור עליהם.

ג.

אחד לאחד:

```
if __name__ == '__main__':
    # Make a collection
    cars: list[str] = []
    cars.append("Volvo")
    cars.append("BMW")
    cars.append("Ford")
    cars.append("Mazda")

    # Get the iterator
    it = iter(cars)

    # Print the first item
    print(next(it))
```

וקצת יותר פייתוני:

```
if __name__ == '__main__':
    # Make a collection
    cars = ["Volvo", "BMW", "Ford", "Mazda"]

    # Get the iterator
    it = iter(cars)

    # Print the first item
    print(next(it))
```

ד. בג'אבה הפונקציונליות נחשבת יותר אדוקה לכללי OOP, כל אובייקט מספק את הפונקציונליות של עצמו לפי הממשקים שהוא מממש. בפייתון אנחנו ניגשים מבחוץ ע"י פונקציות built-in אשר מקבלות אובייקטים שמתאימים לתבניות שלהם. לכן בג'אבה יכולתי לקרוא לאיטרטור של CARS רק בגלל שARRAYLIST מממש ITERABLE, אחרת לא הייתי יכול, בפייתון הייתי יכול לקרוא לITER על CARS גם אם הסוג של CARS לא היה מממש, ואז מייד התוכנית הייתה עפה על TypeError...
זה מקרה פרטי של העניין הרחב מאוד שלפונקציונליות כזו ודומות לה שהקריאה להם מובנית בפייתון לא מממשים ממשקים אלא "Magic Methods", לדוגמא __repr__, __hash__, __gt__, __eq__, __add__ וכו'