

הנחיות כלליות:

- מועד אחרון להגשה: כמפורסם בתיבת ההגשה ב-Moodle.
- מטרת התרגיל הינה לתרגל כתיבה, ולרכוש מיומנות בכלים שנלמדו עד כה בכיתה.
- קראו את העבודה מתחילתה ועד סופה לפני שאתם מתחילים לפתור אותה. ודאו שאתם מבינים את כל המשימות.
- רמת הקושי של המשימות אינה אחידה.
- את התרגיל יש לפתור לבד!
- בתיבת ההגשה במערכת ה-VPL ישנו קובץ שלד לכתובת הקודם שלכם. עבור כל משימה כתבו הפתרון שלכם במקום המתאים למשימה. **בעבודה זאת סופקו לכם 2 קבצי שלד: `company_m.py` ו-`stock_market_m.py`.**
- אין למחוק שום קטע קוד הנמצא בשלד. עליכם רק להוסיף את הפתרון שלכם בתוכו.
- אין לשנות את שם/שמות ה-Requested Files.
- השאלות יבדקו באופן אוטומטי. הפלט שעליכם להחזיר בכל תרגיל צריך להיות בדיוק כפי שנדרש. כמו כן, באופן אקראי יבדקו גם עבודות באופן ידני.
- כאשר תבוצע בדיקה ידנית, תתבצע גם בדיקת Readability - שימו לב שאתם משתמשים בשמות משתנים אינפורמטיביים וכותבים הערות בכל סעיף.
- בדיקה עצמית: כדי לוודא את נכונותן ואת עמידותן של הפונקציות לקליטים שונים, בכל שאלה הריצו אותן עם מגוון קליטים: אלה שמופיעים בדוגמאות וקליטים נוספים עליהם חשבתם. וודאו כי הפלט נכון. הבדיקה תתבצע על מגוון דוגמאות ולא בהכרח אלה שיינתנו פה.
- ניתן להשתמש בחומר הנלמד עד לפרסום העבודה ורק בחומר הזה.
- אין להשתמש בחבילות או מודולים חיצוניים (כדוגמת math) למעט מקרים שבהם צוין אחרת במפורש.
- במידה ולא צוין אחרת, **אין** להניח את נכונות הקלט.
- משקל כל שאלה הוא זהה.
- במידה ולא עניתם על סעיף מסוים, נא מלאו את הפונקציה/שיטה בכל מקרה על מנת שהקוד שלכם יצליח לעבוד.

הערות מיוחדות:

בסוף העבודה מוצג קוד דוגמה, והפלט הרצוי שמהרצה של קוד דוגמה זה. כמו כן, קוד הדוגמה יינתן לכם כקובץ `main.py`. תוכלו לעשות בו שימוש אישי, אך אין להגיש אותו בתיבת ההגשה.

תיאור המטלה

במטלה זו עליכם ליצור מערכת המדמה את שוק ההון. שוק הון (Stock Market) מורכב ממספר חברות (Company). המערכת תאפשר למשתמש להכניס ולהסיר חברות משוק ההון ולבצע פעולות שונות. אתם תבנו מחלקת Company שתכיל שיטות נדרשות ותייצג חברה בודדת. כמו כן, אתם תבנו מחלקת StockMarket שתייצג את שוק ההון כולו ותכיל רשימה של חברות.

הערה: במטלה זאת נעשה שימוש במילה שדה על מנת לתאר instance attribute.

המחלקה Company (50%)

המחלקה Company מייצגת חברה אשר נסחרת בבורסה. להלן השדות שיש להגדיר במחלקה:

name – מסוג string, ומייצג את שם החברה. שם החברה יכול להיות מורכב ממילים (באנגלית) ורווחים המפרידים ביניהם (רווח בודד בין 2 מילים). מילה היא רצף של לכל הפחות 2 אותיות באנגלית. כמו כן, על שם החברה להתחיל באות גדולה באנגלית (למשל: "NVIDIA Corporation").

stocks_num – מסוג int, ומייצג את מספר המניות שיש לחברה. על מספר המניות להיות חיובי.

stock_price – מסוג float או int, ומייצג מחיר מנייה בודדת. על מחיר מנייה להיות חיובי.

comp_type – מסוג string, ומייצג את סוג החברה. על המחרוזת לקיים את אותם התנאים שמקיימת המחרוזת של השדה name.

אין לשנות את השדות ואין להוסיף שדות נוספים.

שיטות נדרשות למימוש במחלקה זו:

תיאור	שיטה
בנאי זה מקבל שם חברה, מספר המניות שלה, מחיר מנייה, וסוג החברה, ויוצר חברה חדשה עם הערכים המתאימים. במידה והתקבל ערך שאינו עונה להגדרות השדות לעיל, על התכנית לזרוק שגיאה מסוג ValueError עם הודעה כרצונכם. שימו לב שאין להניח הנחות הקלטים בבנאי.	<code>__init__(self, name, stocks_num, stock_price, comp_type)</code>
השיטה מחשבת ומחזירה את שווי החברה. שווי החברה מחושב לפי: מחיר למניה כפול מספר המניות.	<code>net_worth(self)</code>
השיטה מעדכנת את שם החברה, בתנאי שהשם עומד בתנאי השדה <code>name</code> לעיל. במידה והיה עדכון, על השיטה להחזיר את הערך הבוליאני True. אחרת, היא לא מעדכנת את השם ומחזירה False.	<code>set_name(self, name)</code>
השיטה מעדכנת את מספר המניות של החברה, בתנאי שהמספר עומד בתנאי השדה <code>stocks_num</code> לעיל. במידה והיה עדכון, על השיטה להחזיר את הערך הבוליאני True. אחרת, היא לא מעדכנת את מספר המניות ומחזירה False. שימו לב: שיטה זאת אינה משנה את ערך שווי החברה. לכן יש להקטין או להגדיל את ערך המנייה של החברה כך שהשווי הכללי יישאר כפי שהיה לפני העדכון.	<code>set_stocks_num(self, stocks_num)</code>
השיטה מעדכנת את מחיר מנייה בודדת של החברה, בתנאי שהמספר עומד בתנאי השדה <code>stock_price</code> לעיל. במידה והיה עדכון, על השיטה להחזיר את הערך הבוליאני True.	<code>set_stock_price(self, stock_price)</code>

	<p>אחרת, היא לא מעדכנת את מחיר המנייה ומחזירה False. שימו לב: שיטה זאת יכולה לשנות את ערך שווי החברה. עליכם לשנות את מספר המניות להיות מקסימלי במובן כזה ששווי החברה לאחר עדכון מחיר המנייה לא יעלה על שווי החברה לפני עדכון מחיר המנייה. זכרו שמספר המניות צריך להיות מספר שלם חיובי. במידה והשיטה קיבלה מחיר מנייה חדש הגדול מערך שווי החברה (לפני העדכון), עליכם להחזיר False ולא לבצע שום עדכון.</p>
set_comp_type(self, comp_type)	<p>השיטה מעדכנת את סוג החברה, בתנאי שהסוג עומד בתנאי השדה comp_type לעיל. במידה והיה עדכון, על השיטה להחזיר את הערך הבוליאני True. אחרת, היא לא מעדכנת את הסוג ומחזירה False.</p>
update_net_worth(self, net_worth)	<p>השיטה מעדכנת את שווי החברה כך שמספר המניות נשאר זהה ומחיר מנייה בודדת משתנה. על ערך net_worth להיות חיובי. במידה והיה עדכון, על השיטה להחזיר את ערך הבוליאני True. אחרת (ז"א במידה ו-net_worth ערך לא תקין), על השיטה לא לשנות דבר ולהחזיר False. ניתן להניח כי net_worth המתקבל הוא מספר.</p>
add_stocks(self, number)	<p>השיטה מקבלת מספר שלם ומוסיפה מספר זה למספר המניות הקיים. שימו לב, המספר יכול להיות שלילי ועל כן יש לחסר. במידה ו-number אינו שלם או שמספר המניות הופך להיות שלילי/אפס, מספר המניות לא מתעדכן הערך המוחזר מן השיטה הוא הערך הבוליאני False. אחרת, יש לעדכן את מספר המניות בהתאם, ולהחזיר True.</p> <p>(דוגמא: במידה ויש 5 מניות בחברה, עבור number=-8 יש לחסר 8 מניות ממספר המניות הקיים. על כן, לא יתכן שיהיו 3- מניות, ולכן אין לעדכן את מספר המניות ויש להחזיר False).</p> <p>ניתן להניח כי number הוא מספר שלם.</p>
__repr__(self)	<p>השיטה מחזירה String אשר מכיל פירוט אודות החברה. לדוגמה עבור חברה בשם "NVIDIA Corporation" מסוג "High tech" בעלת 1000 מניות כאשר מחיר מנייה הוא 20.284, תוחזר המחרוזת הבאה:</p> <p>"NVIDIA Corporation 1000 stocks, Price: 20.284, High Tech, Net Worth: 20284.0"</p> <p>(שימו לב כי מדובר במחרוזת בעלת שורה אחת, ולאחר כל פסיק ישנו רווח. <u>בסוף המחרוזת המחוזרת אין להוסיף "\n"</u>)</p>
__lt__(self, other) __gt__(self, other) __eq__(self, other)	<p>יש לבצע Operator Overloading לאופרטורי ההשוואה <, >, == כך שההשוואה תהיה על סמך שווי החברה שהוגדר לעיל.</p>

	<p>לדוגמה, עבור חברות A ו-B שהשווי שלהם הוא 300.5 ו-405.3, בהתאמה, אם נחשב את הביטוי $B > A$ אז נקבל True שכן שווי החברה B גדול משווי חברה A.</p>
<code>__add__(self, other)</code>	<p>יש לבצע Operator Overloading לאופרטור החיבור + כך שפעולת החיבור תמזג את החברות שעליהם נבצע פעולת החיבור. האופרנד הימני לאופרנד השמאלי (קרי, החברה מימין לחיבור תתמזג לחברה משמאל לחיבור. משיטה זאת תחזור חברה חדשה שתהווה החברה הממוזגת. המיזוג של החברות יתבטא כך: מספר המניות של החברה החדשה יהיה סכום מספרי המניות של החברות. המחיר למניה של החברה החדשה יחושב לפי סכום השווי של שתי החברות (net worth) לחלק למספר המניות החדש. שם וסוג החברה יהיו כשם וסוג החברה משמאל לפעולת החיבור (ז"א האופרנד השמאלי). שימו לב: השיטה מחזירה חברה חדשה (כלומר instance חדש מסוג company), ואין לשנות את החברות הקיימות.</p>

המחלקה StockMarket (50%)

המחלקה `StockMarket` מכילה רשימת חברות אשר שייכות לשוק. היא מכילה שיטות שונות אשר מאפשרות לקבל מידע אודות החברות ולבצע שינויים. להלן השדות שיש להגדיר במחלקה:

`name` – מסוג `string`, ומייצג את שם שוק ההון. על השם להכיל אך ורק אותיות באנגלית (לכל הפחות 2), ולהתחיל באות גדולה. (למשל: "Nasdaq").

`min_net_worth_threshold` – מסוג `int` ומהווה סף שווי חברה מינימלי לכניסה לשוק ההון. ערך זה חייב להיות אי-שלילי.

`stock_market_types` – רשימה של `strings`. הרשימה מייצגת את סוגי החברות שיכולות להיכנס לשוק ההון. על השמות הסוגים לקיים את התנאי על השמות שהוגדר במחלקה `Company`.

`companies` – רשימה שאבריה הם מסוג `Company`. הרשימה מייצגת את החברות השונות שנסחרות בשוק ההון. לכל חברה בשוק ההון חייב להיות שם ייחודי.

הערה: מרגע שחברה נמצאת בשוק ההון, לא קיימת בקרה המוודא ששווי החברה אינו יורד מתחת ל-`min_net_worth_threshold` או שסוג החברה לא השתנה לסוג שאינו נמצא ב-`stock_market_types`.

אין לשנות את השדות ואין להוסיף שדות נוספים.

שיטות נדרשות למימוש במחלקה זו:

תיאור	שיטה
<p>בנאי זה מקבל שם שוק ההון, סף שווי כניסה, רשימת סוגי חברות שיכולות להיכנס לשוק ההון, ורשימה של חברות שיכנסו לשוק ההון. שימו לב לערך ה-default של <code>companies</code>. משמעות הערך <code>None</code> היא לייצור שוק הון ללא חברות כלל (כלומר ערך השדה <code>companies</code> יהיה רשימה ריקה). לכן, אם לא מכניסים ארגומנט זה בקריאת הבנאי, אזי שוק ההון</p>	<p><code>__init__(self, name, min_net_worth_threshold, stock_market_types, companies=None)</code></p>

	<p>יהיה ללא חברות כלל. במידה והתקבל ערך שאינו עונה להגדרות השדות לעיל, על התכנית לזרוק שגיאה מסוג ValueError עם הודעה כרצונכם. שימו לב: במידה והרשימה companies אינה ריקה, עליכם להכניס לשוק ההון רק את החברות שעומדות בתנאים לכניסה לשוק (מבחינת שווי מינימום, וסוג החברה). במידה והרשימה companies תקינה (ז"א מכילה רק משתנים מסוג company), אזי ניתן להניח שהרשימה companies אינה מכילה 2 חברות בעלות שם זהה. מעבר לכך אין להניח דבר על הקלטים לבנאי.</p>
market_net_worth(comp_type)	<p>השיטה מחזירה את הסכום של שווי (net worth) כל החברות מסוג comp_type שבשוק. אם אין בשוק חברות מסוג זה, השיטה מחזירה 0. ניתן להניח כי comp_type הוא מסוג מחרוזת. אין צורך לבדוק את תקינות comp_type.</p>
insert(self, c)	<p>השיטה מקבלת חברה c ומוסיפה אותה לרשימת החברות של השוק. אם החברה כבר קיימת בשוק (כלומר קיימת בשוק ההון חברה עם שם זהה) או שהחברה לא עומדת בתנאי הכניסה לשוק (כלומר לא עומדת בסף שווי החברה או שסוג החברה אינו תואם לסוגי החברות המותרות בשוק), השיטה לא מכניסה את החברה c, ומחזירה את הערך הבוליאני False. אחרת, על השיטה להכניס את החברה לשוק ההון (קרי, לרשימת החברות בשוק), ולהחזיר True. ניתן להניח כי c הוא מסוג company.</p>
top_n(self, n)	<p>השיטה מחזירה רשימה ממוינת של n החברות בשוק ההון שווי החברה שלהן הוא הגדול ביותר. על הרשימה להיות ממוינת מהחברה עם השווי הגדול ביותר לחברה עם השווי הקטן ביותר ברשימה זו. במידה ו-n גדול מכמות החברות בשוק ההון, על הרשימה הממוינת להכיל את כל החברות בשוק ההון. במידה ואין חברות בשוק ההון, על הרשימה המוחזרת להיות ריקה. ניתן להניח כי n הוא מספר שלם חיובי.</p>
merger(self, c1, c2)	<p>השיטה מקבלת 2 חברות וממזגת ביניהן. כיצד: השיטה תיצור חברה חדשה בעזרת פעולת המיזוג הקיימת במחלקה company. על שם וסוג החברה להיות כשם וסוג החברה בעלת שווי גדול יותר. במידה והחברות בעלות שווי זהה, על שם וסוג החברה להיות כשם וסוג החברה של c1. השיטה תוסיף לרשימת החברות בשוק את החברה חדשה (חברת המיזוג). השיטה תמחק את 2 החברות שהתמזגו מרשימת החברות.</p>

	<p>שימו לב: המיזוג יכול להתבצע רק אם שתי החברות מופיעות ברשימת החברות של השוק. במידה והמיזוג הצליח, על השיטה להחזיר את החברה החדשה. אחרת, השיטה תחזיר None. ניתן להניח כי c1 ו-c2 הם מסוג company.</p>
--	---

קוד לדוגמה:

```

from company_m import Company
from stockMarket_m import StockMarket

def main():
    print("#####Company tests#####")
    print("1. create Company instance:")
    c1 = Company("Google", 1000, 20.284, "High Tech")
    print(c1)

    print("2. test net_worth method:")
    print(c1.net_worth())

    print("3. test set_name method:")
    print(c1.set_name("Google2"))
    print(c1.set_name("Google Two"))
    print(c1)

    print("4. test set_stocks_num method:")
    print(c1.set_stocks_num(2000))
    print(c1)

    print("5. test set_stock_price method:")
    print(c1.set_stock_price(25))
    print(c1)

    print("6. test set_comp_type method:")
    print(c1.set_comp_type("General"))
    print(c1)

    print("7. test update_net_worth method:")
    print(c1.update_net_worth(0))
    print(c1)
    print(c1.update_net_worth(2027.5))
    print(c1)

    print("8. test add_stocks method:")
    print(c1.add_stocks(-850))
    print(c1)
    print(c1.add_stocks(8000))
    print(c1)

    print("9. test Operator Overloading:")
    c2 = Company("Lenovo", 1000, 5, "High Tech")
    print(c2)
    print(c1 > c2)
    print(c1 < c2)
    print(c1 == c2)
    print(c1 + c2)

    print("#####StockMarket tests#####")
    c1 = Company("Google", 1000, 20.284, "High Tech")
    c2 = Company("Lenovo", 1000, 5, "High Tech")

```

```

c3 = Company("Pfizer", 4000, 10, "Pharmaceutical")
c4 = Company("Apple", 5000, 10.4, "Hardware")

print("1. create StockMarket instance:")
stock_market = StockMarket("Nasdaq", 6000, ["High Tech",
"Hardware"], companies=[c1, c2, c3, c4])
for company in stock_market.companies:
    print(company)

print("2. test insert method:")
c5 = Company("Microsoft", 5000, 15.565, "High Tech")
print(stock_market.insert(c5))
for company in stock_market.companies:
    print(company)

print("3. test market_net_worth method:")
print(stock_market.market_net_worth("High Tech"))
print(stock_market.market_net_worth("Pharmaceutical"))

print("4. test top_n method:")
top_n_companies = stock_market.top_n(2)
for company in top_n_companies:
    print(company)

print("5. test merger method:")
new_c = stock_market.merger(c4, c5)
print(new_c)
for company in stock_market.companies:
    print(company)

if __name__ == "__main__":
    main()

```

הפלט הרצוי לקוד זה:

#####Company tests#####

1. create Company instance:

Google, 1000 stocks, Price: 20.284, High Tech, Net Worth: 20284.0

2. test net_worth method:

20284.0

3. test set_name method:

False

True

Google Two, 1000 stocks, Price: 20.284, High Tech, Net Worth: 20284.0

4. test set_stocks_num method:

True

Google Two, 2000 stocks, Price: 10.142, High Tech, Net Worth: 20284.0

5. test set_stock_price method:

True

Google Two, 811 stocks, Price: 25, High Tech, Net Worth: 20275

6. test set_comp_type method:

True

Google Two, 811 stocks, Price: 25, General, Net Worth: 20275

7. test update_net_worth method:

False

Google Two, 811 stocks, Price: 25, General, Net Worth: 20275

True

Google Two, 811 stocks, Price: 2.5, General, Net Worth: 2027.5

8. test add_stocks method:

False

Google Two, 811 stocks, Price: 2.5, General, Net Worth: 2027.5

True

Google Two, 8811 stocks, Price: 2.5, General, Net Worth: 22027.5

9. test Operator Overloading:

Lenovo, 1000 stocks, Price: 5, High Tech, Net Worth: 5000

True

False

False

Google Two, 9811 stocks, Price: 2.7548160228315157, General, Net Worth: 27027.5

#####StockMarket tests#####

1. create StockMarket instance:

Google, 1000 stocks, Price: 20.284, High Tech, Net Worth: 20284.0

Apple, 5000 stocks, Price: 10.4, Hardware, Net Worth: 52000.0

2. test insert method:

True

Google, 1000 stocks, Price: 20.284, High Tech, Net Worth: 20284.0

Apple, 5000 stocks, Price: 10.4, Hardware, Net Worth: 52000.0

Microsoft, 5000 stocks, Price: 15.565, High Tech, Net Worth: 77825.0

3. test market_net_worth method:

98109.0

0

4. test top_n method:

Microsoft, 5000 stocks, Price: 15.565, High Tech, Net Worth: 77825.0

Apple, 5000 stocks, Price: 10.4, Hardware, Net Worth: 52000.0

5. test merger method:

Microsoft, 10000 stocks, Price: 12.9825, High Tech, Net Worth: 129825.0

Google, 1000 stocks, Price: 20.284, High Tech, Net Worth: 20284.0

Microsoft, 10000 stocks, Price: 12.9825, High Tech, Net Worth: 129825.0