

מדור בחינות ומערכת שעות

היחידה למכינה

—/—/—

00:00-00:00

צעדים ראשונים בתכנות

מועד demo רמת 3 מתוך 5

סויה דורון

תשפ"ה סמסטר א'

השאלון מכיל 00 עמודים כולל עמוד זה

בהצלחה !

=====

הנחיות למדור בחינות

שאלוני בחינה

- ☒ לשאלון הבחינה אין לצרף מחברת
- ☒ יש להחזיר את השאלון.
- ☒ הפתרון הוא על השאלון בחינה עצמה **בלבד**.

הנחיות כללי

שימוש במחשבוני

- ☒ ניתן להשתמש במחשבון
- ☐ לא ניתן להשתמש במחשבון

חומר עזר

- ☒ ניתן להשתמש בחומר עזר/דף נוסחאות של **מרצה בלבד**.

אחר / הערות

שאלה מס' 1 - 35 נק':

שימו לב יש לקרוא את כל השאלה לפי תחילת פתרון!
סעיף א - 20 נק': כתבו את המימוש של מערכת לניהול חנות מחשבים.

מחלקה אבסטרקטית Employee

תכונות:

- name: שם העובד.
- id: מספר מזהה של העובד.
- department: המחלקה בה הוא עובד (כגון "מכירות", "טכני", "שירות")

מתודות:

get_salary(self) המתודה אמורה להחזיר את השכר של העובד.
get_details(self) מחזירה מחרוזת הכוללת: סוג העובד, שם, מזהה, מחלקה, שכר

```
from abc import ABC, abstractmethod
```

```
class Employee(ABC):
```

```
    def __init__(self, name, id, department):
```

CommissionEmployee

תכונות:

- `sales_amount`: סך המכירות שביצע החודש
- `commission_rate`: אחוז עמלה (למשל 0.1 עבור 10% מהמכירות שלו)

מתודות:

`get_salary(self)` מחזירה את השכר.

`get_details(self)` המתודה מחזירה מחרוזת המתארת את העובד, כל מה שאפשר לספר עליו.

```
class CommissionEmployee(_____):
    def __init__(self, name, id, department, sales_amount,
commission_rate):
```

```
def get_salary(self):
```

```
def get_details(self):
```

Technician

תכונות:

- `hourly_rate`: תעריף לשעה
- `tasks_completed`: מילון (dict) שבו כל מפתח הוא משימה (str) והערך הוא משך הזמן בביצוע (int, בדקות)

מתודות:

`get_salary(self)` מחזירה את השכר סה"כ דקות / 60 * תעריף לשעה, אם אחת המשימות ארוכה מ-90 דקות, יש להוסיף בונוס חד-פעמי של 100 ש"ח לשכר.
`get_details(self)` המתודה מחזירה מחרוזת המתארת את העובד כל מה שאפשר לספר עליו.

```
class Technician(_____):
    def __init__(self, name, id, department, hourly_rate,
tasks_completed):
```

```
def get_salary(self):
```

```
def get_details(self):
```

מחלקת ComputerStore

תכונות:

- name: שם החנות
- employees: רשימה של כל העובדים

מתודות:

מתודות:

–add_employee(employee) מוסיפה עובד

–total_salaries() מחזירה סכום כל השכר

–get_store_details() מחזירה מחרוזת עם שם החנות ופרטי כל העובדים

–get_employee_by_id(id) מחזירה את העובד לפי מזהה. אם לא נמצא – זורקת שגיאה

"EmployeeNotFound"

– remove_lowest_paid_employee – מסירה את העובד עם השכר הנמוך ביותר. אם יש כמה –

תסיר את הראשון

```
class ComputerStore:
```

```
    def __init__(self, name):
```

```
    def add_employee(self, employee):
```

```
    def total_salaries(self):
```

```
    def get_company_details(self):
```

```
    def get_employee_by_id(self, id):
```

```
def remove_lowest_paid_employee(self):
```

סעיף ב - 5 נק': מה הפלט של הקוד הבא?

```
e1 = CommissionEmployee("Tomer", 1, "Sales", 30000, 0.12)
e2 = Technician("Dana", 2, "Technical", 80, {"Install OS": 50,
"Upgrade RAM": 120})

shop = ComputerStore("ByteWorld")
shop.add_employee(e1)
shop.add_employee(e2)

print(shop.get_store_details())
print("Total payroll:", shop.total_salaries())
print(shop.get_employee_by_id(2))
```

שאלה מס' 2 - 25 נק': שרטוט בנושא מיונים.

קיבלנו מערך (Array), פתרו אותו בעזרת **Bubble Sort**, שרטטו כל צעד וצעד, **אין לקפוץ על שלבים גם אלמנטרים זה יגרור הורדה בניקוד.**
שימו לב, עליכם לתאר בכמה מילים איך עובד האלגוריתם.

38	27	43	3	9	82	10
0	1	2	3	4	5	6

שאלה מס' 3 - 25 נק': ממש קוד של מבנה נתונים.

סעיף א - 10 נק': ממשו מחלקה בשם **ReverseStack**, המדמה מחסנית שבה הפריט החדש נכנס לתחתית ולא לראש המחסנית. השתמשו רק בטאפלים, ללא שימוש ברשימות או מודולים חיצוניים.

```
class Stack:
```

```
    def __init__(self):
```

```
    def is_empty(self):
```

```
    def push(self, item):
```

```
    def pop(self):
```

```
    def peek(self):
```

```
    def size(self):
```


סעיף ב - 15 נק': ממשו את הפונקציה נוספת.

ממשו פונקציה בשם `copy_stack(stack)` אשר מקבלת אובייקט מטיפוס `ReverseStack` ומחזירה עותק זהה של המחסנית — מבלי לפגוע במקור אין להשתמש בתכונות של רשימה או טפאל.

```
def copy_stack(stack1):
```

שאלה מס' 4 - 15 נק': שאלות נכון/לא נכון

הדרכה:

על כל שאלה יש לענות **נכון** או **לא נכון**,
בשאלות בהם בחרתם **נכון** אין חובת נימוק,
אך בשאלות בהם בחרתם **ב"לא נכון"** עליכם לנמק.
יש לענות בקצר.

1. בפייתון ניתן להמיר מחרוזת למספר שלם בעזרת `int("42")`

2. ניתן להשתמש במילון (dict) כאשר המפתחות הם מסוג tuple.

3. המילון בפייתון שומר תמיד את סדר ההוספה של המפתחות.

4. הקריאה `range(5)` יוצרת רשימה של 5 מספרים

5. ניתן להשתמש ב `else` לאחר לולאת `for`.

שאלה מס' 5 - 10 נק': מושגים בפייתון.
הסבר בקצרה (ב-2-3 שורות) את המשמעות והשימוש של כל אחד מהמושגים הבאים:
יש לענות בעברית, בשפה פשוטה וברורה. כל הסבר שווה 2 נקודות.

1. פונקציה רקורסיבית (Recursive Function)

2. משתנה מקומי (Local Variable)

3. חריגה מסוג TypeError

4. פעולת break בתוך לולאה

5. ההבדל בין append לextend ברשימות
