

## שאלה 1

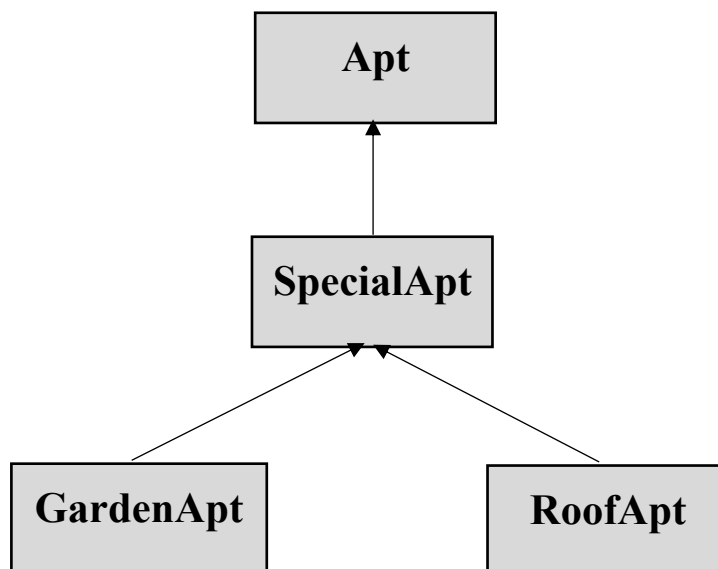
מעוניינים למחשב מערכת מידע של חברת בנייה מובילה.

חברת בנייה בונה בניין בו דירות ממספר סוגים :

- **דירה** (Apt) – מאופיינת על ידי מספר קומה (floor), שטח הדירה במטרים רבועים (area).
- **דירה מיוחדת** (SpecialApt) – היא דירה ומאופיינת בנוסף על ידי האם יש לדירה נוף (has\_view).
- **דירת גן** (GardenApt) – היא דירה מיוחדת ומאופיינת בנוסף על ידי שטח הגינה במטרים רבועים (garden\_area). דירת גן נמצאת תמיד בקומת הקרקע (קומת אפס) שאין בה נוף.
- **דירת גג** (RoofApt) – היא דירה מיוחדת ומאופיינת בנוסף על ידי האם לדירה יש בריכה (has\_pool). לדירת גג יש תמיד נוף.

**שימו לב!** משתני המופע בכל מחלקה **לא יגזרו כציבוריים אלא כמוגנים (protected)** באמצעות מקף תחתון, כלומר self.\_attribute\_name.

עץ הירושה של המחלקות נראה כך :



א. עבור כל אחת מארבעת המחלקות המתוארות לעיל:

- (1) הגדירו כותרת מחלקה (במידה ונדרש, יש להגדיר גם מחלקה יורשת).
- (2) הגדירו פונקציית בנאי מתאימה. במידת הצורך, הוסיפו קבועים (constants), באותיות גדולות, היכן שנדרש.
- (3) הניחו כי הערכים המתקבלים בפונקציות הבנאי תקינים.  
על חתימות הבנאים של המחלקות להיות כדלהלן בדיוק  
עבור המחלקה Apt: `def __init__(self, floor, area)`  
עבור המחלקה SpecialApt: `def __init__(self, floor, area, has_view)`  
עבור המחלקה GardenApt: `def __init__(self, area, garden_area)`  
עבור המחלקה RoofApt: `def __init__(self, floor, area, has_pool)`
- (4) פונקציית `get` עבור כל תכונה. מוסכמת שם הפונקציה היא `get_attribute_name()`.  
לדוגמא, במחלקה Apt תהייה בין היתר פונקציה שחתימתה `def get_floor(self)` עבור התכונה `floor`.
- (5) פונקציית `__eq__` המשווה בין דירה נוכחית לבין דירה המתקבלת כפרמטר. הפונקציה תחזיר `True` במידה וערכי כלל התכונות זהות בין שתי הדירות. אחרת, יוחזר `False`. בהתאם לעקרונות מונחה עצמים והורשה, הקפידו להשתמש בפונקציות `__eq__` יורשות רלוונטיות. במידה ומתקבל אובייקט שאינו מהטיפוס המתאים, יש להחזיר `NotImplemented`.
- (6) פונקציית `__str__` המחזירה מחרוזת המתארת את האובייקט הנוכחי במבנה:  
`attribute_name: attribute_value, attribute_name: attribute_value, ...`  
שימוש לב לרווחים הקיימים (אחרי התו `:` ואחרי התו `,`). בהתאם לעקרונות מונחה עצמים והורשה, הקפידו להשתמש בפונקציות `__str__` יורשות רלוונטיות.

מחירי הדירות נקבעים על פי הכללים הבאים:

- המחיר למטר מרובע אחד הוא 20000 ₪. כל עלייה בקומה מוסיפה למחיר הדירה 5000 ₪ (אין תוספת תשלום לדירות בקומה הראשונה).
- נוף מוסיף למחיר הדירה 600 ₪ X מספר הקומה – למשל, למחיר דירה בעלת נוף הנמצאת בקומה עשירית יש להוסיף 6000 ₪.

- דירת גג מוסיפה למחיר הדירה עוד 40000 ₪ וברכיכה מוסיפה למחיר הדירה 30000 ₪.
  - ב. הוסיפו את הפונקציה `get_price` בכל המחלקות הרלוונטיות כדי לבצע את הנדרש (רק במחלקות שיש בהן צורך), לפי עקרונות תכנות מונחה עצמים.
- הערה: אין להוסיף תכונות/ות למחלקות.

הוסיפו באותו פרויקט קובץ בשם `mmn15.py` עם הפונקציות הבאות:

- ג. כתבו פונקציה בשם `average_price` המקבלת רשימה של דירות, `apts`. הפונקציה תחזיר את ממוצע עלות הדירות.
- ד. כתבו פונקציה בשם `how_many_rooftop` המקבלת רשימה של דירות, `apts`. הפונקציה תחזיר את מספר דירות גג עם בריכה הקיימות ברשימה. במקרה ואין דירות מתאימות, יוחזר הערך 0.
- ה. כתבו פונקציה בשם `how_many_apt_type` המקבלת רשימה של דירות, `apts`. הפונקציה תחזיר מילון (`dict`) בו המפתח יהיה סוג הדירה והערך יהיה כמות הדירות מסוג זה. אלו ערכי המפתחות: `Apt`, `SpecialApt`, `RootApt`, `GardenApt`. כל דירה יכולה להיות מסווגת לסוג דירה אחד בלבד. במקרה והרשימה `apts` ריקה, יוחזר מילון בו כל ערכי המפתחות הוא 0.
- ו. כתבו פונקציה בשם `top_price` המקבלת רשימה של דירות, `apts`. הפונקציה תחזיר את הדירה שעלותה מקסימלית. במידה ויש יותר מדירה אחת כזו, יש להחזיר את הראשונה שנמצאה. במידה והרשימה ריקה, יוחזר `None`.
- ז. כתבו פונקציה בשם `only_valid_apts` המקבלת רשימה של דירות, `apts`. הפונקציה תחזיר רשימה של דירות עם נוף או עם בריכה, שמחירן מעל 1,000,000 שקל. אם אין אף דירה העונה על הדרישות, יש להחזיר `None`.

הערה: בכל הפונקציות בסעיפים ג-ז ניתן להניח כי הרשימה `apts` מייצגת אובייקט מטיפוס רשימה (`list`) וכל איברי הרשימה הם דירות (אובייקטים מטיפוס המחלקה `Apt` או אחת מהמחלקות היורשות ממנה).

## שימו לב, בכל שאלות המטלה.

- אסור להוסיף משתנה מופע או מחלקה למחלקות.
- מותר להוסיף שיטות פרטיות אבל לא ציבוריות.
- אין להשתמש במספרים בקוד. יש להוסיף קבועים עבור כל מספר קבוע ולהשתמש בקבוע בקוד מלבד מקרים טריוויאליים (למשל, הערך 0).
- הקפידו לעשות שימוש בכללי תכנות מונחה עצמים והורשה ולכתוב קוד אחיד לכלל המחלקות, עד כמה שניתן. היכן שניתן יש לקרוא לשיטה הנמצאת במחלקת האב כדי לא לשכפל קוד שלא לצורך.
- אתם צריכים לכתוב בעצמכם docstring למחלקה, לבנאי ולשיטות לפי הנהוג בכתיבת docstring. כמו כן, עליכם לתעד בתיעוד פנימי כל מה שדורש הבהרה ואינו פשוט. דוגמאות לתיעוד ניתן למצוא בפרק 1.9 באתר הקורס.

שימו לב ששמנו טסטר לארבעת המחלקות באתר הקורס. חובה שטסטר זה ירוץ ללא שגיאות קומפילציה עם המחלקות שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטרים ירוצו עם המחלקות ללא שגיאות קומפילציה. אם הטסטרים לא ירוצו בגלל שגיאות קומפילציה הציון במטלה יהיה אפס.

## הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. סעיפים א+ב יכתבו עבור כל מחלקה (Apt, SpecialApt, GardenApt, RoofApt) בקובץ py נפרד. המחלקה Apt תיכתב בקובץ apt.py, המחלקה SpecialApt תיכתב בקובץ special\_apr.py, המחלקה GardenApt תיכתב בקובץ garden\_apr.py, המחלקה RoofApt תיכתב בקובץ roof\_apr.py בנוסף, הפונקציות המופיעות בסעיפים ג-ז, יכתבו בקובץ mmn15.py.
3. ארזו את קובץ הפתרון בקובץ zip ( ולא rar ) יחיד ושלחו אותו בלבד.
4. אל תשכחו לשמור את מספר האסמכתא שקיבלתם מהמערכת לאחר ההגשה. אם לא קיבלתם מספר אסמכתא, סימן שההגשה לא התקבלה.
5. שימו לב, אתם יכולים לשלוח שוב ושוב את המטלה במערכת, אם אתם רוצים לתקן משהו בה. כל הגשה דורסת את ההגשה הקודמת. אבל עשו זאת אך ורק עד לתאריך ההגשה. אחרי התאריך, ייחשב לכם כאילו הגשתם באיחור, גם אם ההגשה הראשונה היתה בזמן! כמו כן, אם המנחה הוריד כבר את המטלה שלכם מהמערכת, לא תוכלו לשלוח עותק מעודכן יותר.

## בהצלחה