Cognome e Nome _____

Matricola _____

Programmazione e Algoritmi 2 A.A. 2024/2025

Esonero del 29 novembre 2024

Compito n° 1

Esercizio 1 [8 punti]

Si consideri il seguente codice Python.

```
class A:
    def metodo (self):
        print ("metodo di A")

class B:
    def metodo (self):
        print ("metodo di B")

class C (A, B):
    pass

class D (C, A):
    pass

class E (B, D):
    pass
```

```
class F (B, A):
    pass

c = C()
c.metodo()

d = D()
d.metodo()

e = E()
e.metodo()

f = F()
f.metodo()
```

Denotando la classe `object` con O, si calcoli il MRO di tutte le classi, e si scriva in tabella il procedimento e il risultato finale:

classe	MRO
A	L(A) = A + merge (L(O), O) = A + merge (O, O) = AO
B	
C	
D	
E	
F	

Se qualcuno di questi MRO non esiste, si eliminino idealmente le linee di codice relativo (sia della definizione che della successiva creazione di oggetti e invocazione di metodo) e si scriva nel riquadro a destra cosa viene stampato a video dal codice restante.

Esercizio 2 [10 punti]

Mostrare **passo-passo l'esecuzione dell'heap-sort** per ordinare l'array **[7, -3, 2, -9, 11, 3, 10, 1, 20]** in modo non decrescente (compresa la fase iniziale di **build-max-heap**).

Esercizio 3 [12 punti]

Si implementino in Python le classi **Articolo** e **Deposito**.
Attenzione: Tutto il codice deve essere annotato con i tipi opportuni.

[4 punti] La *classe* **Articolo** ha i seguenti attributi di istanza:

- **_codice** (una **stringa**)
- **_descrizione** (una **stringa**)
- **_quantità** (un **intero**)

ed i seguenti metodi di istanza:

- *__init__* che inizializza un oggetto della *classe* **Articolo** assegnando descrizione, codice e quantità (il parametro quantità ha 1 come valore di default)
- proprietà (`@property`) in lettura per tutti gli attributi, con lo stesso nome (senza _);
- proprietà in scrittura per descrizione e quantità (la quantità non può mai essere <= 0, e in caso uno tenti di modificarla in tal senso viene impostata a 1).
- metodo **__eq__ (self, o:object)** che restituisce **True** se **o** è un articolo con lo stesso codice di self, (indipendentemente dalla descrizione e dalla quantità), **False** altrimenti.
- metodo **__str__ (self)** che restituisce la rappresentazione come stringa di un articolo, nella forma `Articolo con codice x, descrizione y e quantità z`, dove x, y e z rappresentano il codice, la descrizione e la quantità dell'articolo **self**, rispettivamente.

[8 punti] La *classe* **Deposito** ha i seguente attributi:

- **_nome** (tipo **str**)
- **_contenuto** (tipo **list[Articolo]**)

La *classe* **Deposito** ha i seguenti metodi di istanza:

- *__init__* che inizializza un oggetto della *classe* **Deposito** prendendo in input il nome del deposito, creando un deposito vuoto
- proprietà (`@property`) **nome** in sola lettura per l'attributo **_nome**.
- metodo **get_quantita (self, a:Articolo)** che *(sfruttando implicitamente o esplicitamente il metodo **__eq__** della classe Articolo)* restituisce la quantità dell'articolo **a** presente nel deposito.
- metodo ***add_articolo (self, a:Articolo, n:int=1),*** *che (sfruttando implicitamente o esplicitamente il metodo **__eq__** della classe Articolo)*
  - se **n>0**
    - aggiunge l'articolo **a** alla lista **_contenuto**, se nel deposito non è già presente un articolo con lo stesso codice.
    - altrimenti aggiorna la quantità corrispondente all'articolo con lo stesso codice, incrementandola di **n**.
  - se **n<0**, viene diminuita la quantità attuale dell'articolo con lo stesso codice di **a** (se presente), sottraendo **n**. Laddove la quantità attuale dell'articolo con lo stesso codice di **a** è minore o uguale a **n**, tale articolo viene rimosso dalla lista **_contenuto**.

Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **100** minuti.
- Scrivere **subito** nome, cognome, matricola su OGNI FOGLIO (**compreso questo**).
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- È possibile consultare appunti, libri e dispense.
- Qualsiasi strumento elettronico di calcolo o comunicazione (smartphone, smart*, calcolatrici, palmari, computer, etc…) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**
- Mettere in vista sul banco un valido documento di identità.

Cognome e Nome _____

Matricola _____

Programmazione e Algoritmi 2 A.A. 2024/2025

Esonero del 29 novembre 2024

Compito n° 2

Esercizio 1 [8 punti]

Si consideri il seguente codice Python.

```
class A:
    def metodo (self):
        print ("metodo di A")

class B:
    def metodo (self):
        print ("metodo di B")

class C (B, A):
    pass

class D (C, B):
    pass

class E (A, D):
    pass
```

```
class F (A, B):
    pass

c = C()
c.metodo()

d = D()
d.metodo()

e = E()
e.metodo()

f = F()
f.metodo()
```

Denotando la classe `object` con O, si calcoli il MRO di tutte le classi, e si scriva in tabella il procedimento e il risultato finale:

classe	MRO
A	L(A) = A + merge (L(O), O) = A + merge (O, O) = AO
B	
C	
D	
E	
F	

Se qualcuno di questi MRO non esiste, si eliminino idealmente le linee di codice relativo (sia della definizione che della successiva creazione di oggetti e invocazione di metodo) e si scriva nel riquadro a destra cosa viene stampato a video dal codice restante.

Esercizio 2 [10 punti]

Mostrare **passo-passo l'esecuzione dell'heap-sort** per ordinare l'array **[3, -7, -2, -13, 7, -1, 6, -3, 19]** in modo non decrescente (compresa la fase iniziale di **build-max-heap**).

Esercizio 3 [12 punti]

Si implementino in Python le classi **Articolo** e **Deposito**.
Attenzione: Tutto il codice deve essere annotato con i tipi opportuni.

[4 punti] La *classe* **Articolo** ha i seguenti attributi di istanza:

- **_codice** (una **stringa**)
- **_descrizione** (una **stringa**)
- **_quantità** (un **intero**)

ed i seguenti metodi di istanza:

- *__init__* che inizializza un oggetto della *classe* **Articolo** assegnando descrizione, codice e quantità (il parametro quantità ha 1 come valore di default)
- proprietà (`@property`) in lettura per tutti gli attributi, con lo stesso nome (senza _);
- proprietà in scrittura per descrizione e quantità (la quantità non può mai essere <= 0, e in caso uno tenti di modificarla in tal senso viene impostata a 1).
- metodo **__eq__ (self, o:object)** che restituisce **True** se **o** è un articolo con lo stesso codice di self, (indipendentemente dalla descrizione e dalla quantità), **False** altrimenti.
- metodo **__str__ (self)** che restituisce la rappresentazione come stringa di un articolo, nella forma `Articolo con codice x, descrizione y e quantità z`, dove x, y e z rappresentano il codice, la descrizione e la quantità dell'articolo **self**, rispettivamente.

[8 punti] La *classe* **Deposito** ha i seguente attributi:

- **_nome** (tipo **str**)
- **_contenuto** (tipo **list[Articolo]**)

La *classe* **Deposito** ha i seguenti metodi di istanza:

- *__init__* che inizializza un oggetto della *classe* **Deposito** prendendo in input il nome del deposito, creando un deposito vuoto
- proprietà (`@property`) **nome** in sola lettura per l'attributo **_nome**.
- metodo **get_quantita (self, a:Articolo)** che *(sfruttando implicitamente o esplicitamente il metodo **__eq__** della classe Articolo)* restituisce la quantità dell'articolo **a** presente nel deposito.
- metodo ***add_articolo (self, a:Articolo, n:int=1),*** *che (sfruttando implicitamente o esplicitamente il metodo **__eq__** della classe Articolo)*
  - se **n>0**
    - aggiunge l'articolo **a** alla lista **_contenuto**, se nel deposito non è già presente un articolo con lo stesso codice.
    - altrimenti aggiorna la quantità corrispondente all'articolo con lo stesso codice, incrementandola di **n**.
  - se **n<0**, viene diminuita la quantità attuale dell'articolo con lo stesso codice di **a** (se presente), sottraendo **n**. Laddove la quantità attuale dell'articolo con lo stesso codice di **a** è minore o uguale a **n**, tale articolo viene rimosso dalla lista **_contenuto**.

Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **100** minuti.
- Scrivere **subito** nome, cognome, matricola su OGNI FOGLIO (**compreso questo**).
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- È possibile consultare appunti, libri e dispense.
- Qualsiasi strumento elettronico di calcolo o comunicazione (smartphone, smart*, calcolatrici, palmari, computer, etc…) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**
- Mettere in vista sul banco un valido documento di identità.