



Pattern Recognition
and Applications Lab

La Programmazione ad Oggetti in Python

Docente: Ambra Demontis

Anno Accademico: 2024 - 2025

Corso di Laurea in Ingegneria Elettronica, Informatica e delle
Telecomunicazioni



University of Cagliari,
Italy

Department of Electrical and
Electronic Engineering



La Programmazione ad Oggetti in Python

Progettare programmi di media complessità in Python (seguendo il paradigma della programmazione orientata agli oggetti).

Le Fasi di Sviluppo di un Programma

1. Object-Oriented Analysis: si ragiona sul problema da risolvere e si *identificano* gli oggetti, le loro proprietà e le interazioni tra gli stessi

2. Object-Oriented Design: si crea un prototipo, cioè si schematizza l'idea derivante dall'analisi al passo precedente

3. Object-Oriented Programming: si implementa il prototipo nel linguaggio di programmazione orientato agli oggetti scelto

La Programmazione ad Oggetti in Python

Caso di studio 1:

Siete stati assunti da un'azienda che produce prodotti elettronici per realizzare un programma amministrativo.

Il programma deve permettere di memorizzare le vendite effettuate ai negozianti. Per la vendita di ogni prodotto, deve memorizzare le seguenti informazioni: giorno, mese, anno, `idx_prodotto`, quantità, `partita_iva_negoziante`. L'`idx_prodotto` identifica un determinato modello di prodotto.

Deve inoltre permettere di creare un grafico dei ricavi ottenuti in un anno scelto. Il ricavo per la vendita di un'unità di prodotto è pari al prezzo di vendita meno il costo di produzione.

La Programmazione ad Oggetti in Python

Il prezzo di vendita dipende dal modello del prodotto, mentre il costo di produzione dipende dal tipo di prodotto:

Tutti gli smartphone hanno un costo di produzione che dipende dal costo dei componenti elettronici + 500 euro di manodopera.

Tutti i computer hanno un costo di produzione che dipende dal costo dei componenti elettronici + 20% del prezzo del prodotto.

Si suppone che il prezzo sia fisso e non possa variare nel tempo.

Per ora l'azienda vi chiede di considerare solo smartphone e pc come prodotti ma vi specificano che in futuro hanno intenzione di ampliare questo programma per inserire tipi differenti di prodotti, come le smart tv.

La Programmazione ad Oggetti in Python

Capire come modellare al meglio un problema reale in Python può non essere banale.

Non esiste un modo univoco di procedere ma l'**importante** è **procedere** con ordine **seguendo una “strategia”**.

Ad esempio, quella mostrata nelle slide successive.

La Programmazione ad Oggetti in Python

Prepariamo un draft di alto livello.

- 1) Quali **classi** ci servono per modellare il problema e quali sono le **relazioni** tra loro? Può essere utile definire delle **superclassi (eventualmente astratte)**?
- 2) Quali sono i **metodi/attributi che** le classi **devono avere**?
Quali **attributi** devono essere **pubblici** e quali devono essere **privati**?

La Programmazione ad Oggetti in Python

Rifiniamo il draft.

- 1) Ci sono **altri metodi/attributi** che possono esserci **utili**? (Per rendere il programma più efficiente o per scomporre un metodo in sottoproblemi).
- 2) Può essere utile far sì che le istanze di alcune delle classi siano **iterabili**?
- 3) Può essere utile definire **oggetti di tipo eccezione**?

La Programmazione ad Oggetti in Python

Prepariamo un draft di alto livello.

1) Quali **classi** possono servirci per modellare il problema e quali sono le **relazioni** tra loro? Può essere utile definire delle **superclassi (eventualmente astratte)**?

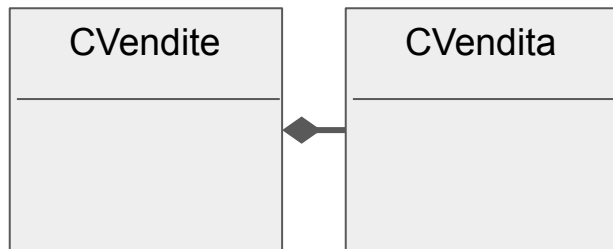
*“Il programma deve permettere di memorizzare le vendite effettuate ai negozianti. **Per la vendita di ogni prodotto**, deve memorizzare le seguenti informazioni: giorno, mese, anno, idx_prodotto, quantità, partita_iva_negoziante.”*

La Programmazione ad Oggetti in Python

Per realizzare questo ci servirà:

- Una classe CVendita che conterrà le informazioni richieste.
- Una classe CVendite che ci permetterà di memorizzare più vendite.

La classe CVendite sarà composta da oggetti di tipo CVendita.



La Programmazione ad Oggetti in Python

“Deve inoltre permettere di creare un grafico dei ricavi ottenuti in un anno scelto. Il ricavo per la vendita di un’unità di prodotto è pari al prezzo di vendita meno il costo di produzione.”

La Programmazione ad Oggetti in Python

*“Deve inoltre **permettere di creare un grafico** dei ricavi ottenuti in un anno scelto. Il ricavo per la vendita di un’unità di prodotto è pari al prezzo di vendita meno il costo di produzione. ”*

La Programmazione ad Oggetti in Python

Per creare il grafico possiamo avere una classe ad-hoc.

Ad esempio CCreatoreGrafici.

(Più classi scegliamo di avere più il codice risulterà manutenibile, e facilmente ampliabile).



La Programmazione ad Oggetti in Python

“Il prezzo di vendita dipende dal modello del prodotto, mentre quello di produzione dipende dal tipo di prodotto:

Tutti gli smartphone hanno un costo di produzione che dipende dal costo dei componenti elettronici + 500 euro di manodopera.

Tutti i computer hanno un costo di produzione che dipende dal costo dei componenti elettronici + 20% del prezzo del prodotto. ”

La Programmazione ad Oggetti in Python

“Il prezzo di vendita dipende dal modello del prodotto, mentre quello di produzione dipende dal tipo di prodotto:

Tutti gli smartphone hanno un costo di produzione che dipende dal costo dei componenti elettronici + 500 euro di manodopera.

Tutti i computer hanno un costo di produzione che dipende dal costo dei componenti elettronici + 20% del prezzo del prodotto. ”

La Programmazione ad Oggetti in Python

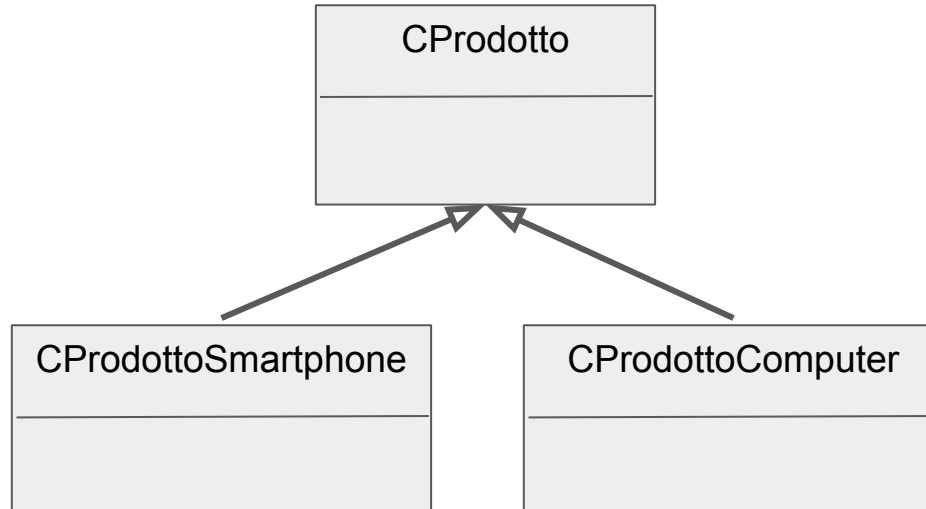
Ci servono delle classi per essere in grado, dato l'idx del prodotto, di ricavare il suo prezzo di vendita e il costo di produzione (che ci servono per calcolare il ricavo).

La Programmazione ad Oggetti in Python

A prescindere dal tipo di prodotto, dobbiamo memorizzare l'idx del **prodotto** (per poterlo cercare) e il **prezzo di vendita**. Ci serve quindi una classe *CProdotto*.

Il **prezzo di produzione** (che va calcolato) è **differente a seconda del tipo di prodotto** (smartphone o computer) dovremo avere due classi. *CProdottoSmartphone* e *CProdottoComputer*.

La Programmazione ad Oggetti in Python



La Programmazione ad Oggetti in Python

“Per ora l’azienda vi chiede di considerare solo smartphone e pc come prodotti ma vi specificano che in futuro hanno intenzione di ampliare questo programma per inserire tipi differenti di prodotti, come le smart tv.”

La Programmazione ad Oggetti in Python

*“Per ora l’azienda vi chiede di considerare solo smartphone e pc come prodotti ma vi specificano che **in futuro hanno intenzione di** ampliare questo programma per **inserire tipi differenti di prodotti**, come le smart tv.”*

La Programmazione ad Oggetti in Python

Come abbiamo visto prima, **i diversi tipi di prodotti devono avere un metodo che permette di calcolare il prezzo di produzione.**

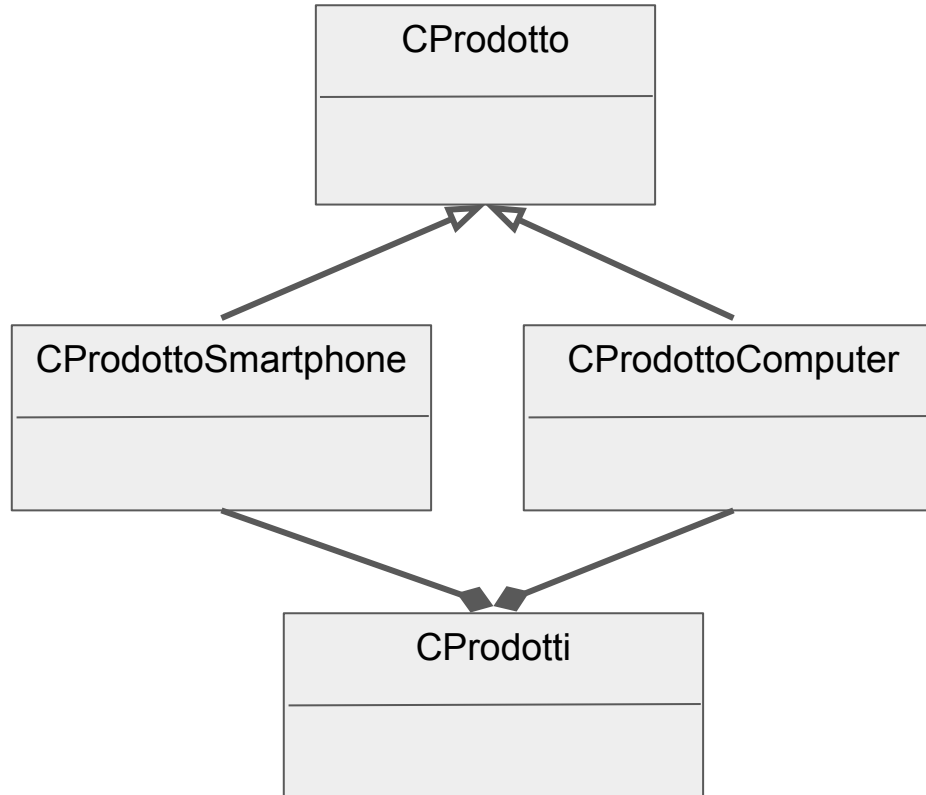
Ovviamente per far sì che il codice funzioni a prescindere dal tipo di prodotto, questo metodo dovrà avere la stessa interfaccia per tutti i tipi di prodotti.

Poichè la gamma di prodotti verrà ampliata in futuro, sarà utile definire CProdotto come classe astratta.

La Programmazione ad Oggetti in Python

Dobbiamo memorizzare **tutti i prodotti** dell'azienda. Per questo possiamo creare una classe ad-hoc: `CProdotti`, che sarà composta da istanze delle classi *CProdottoSmartphone*, *CProdottoComputer*.

La Programmazione ad Oggetti in Python



La Programmazione ad Oggetti in Python

2) Quali sono i **metodi/attributi** che le classi **devono avere**?

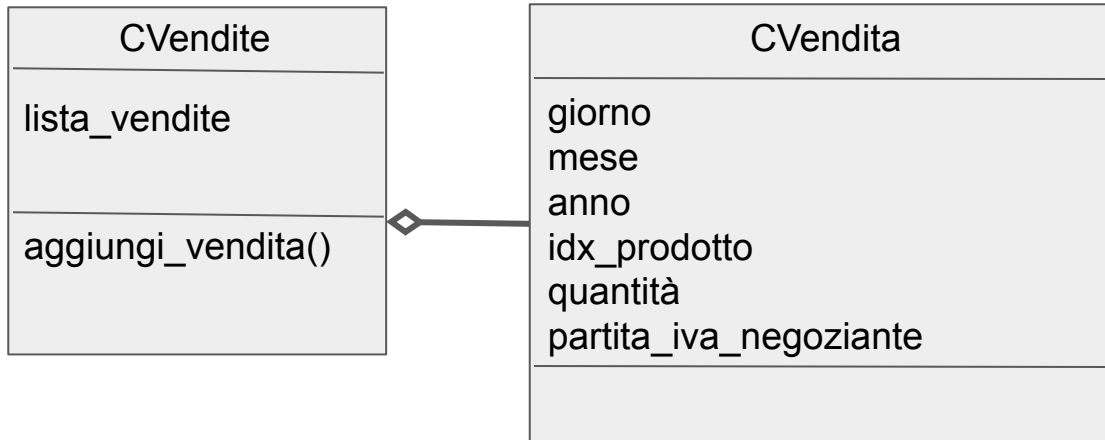
Quali **attributi** devono essere **pubblici** e quali devono essere **privati**?

Analizziamo i vari diagrammi delle classi e annotiamo attributi metodi indispensabili.

La Programmazione ad Oggetti in Python

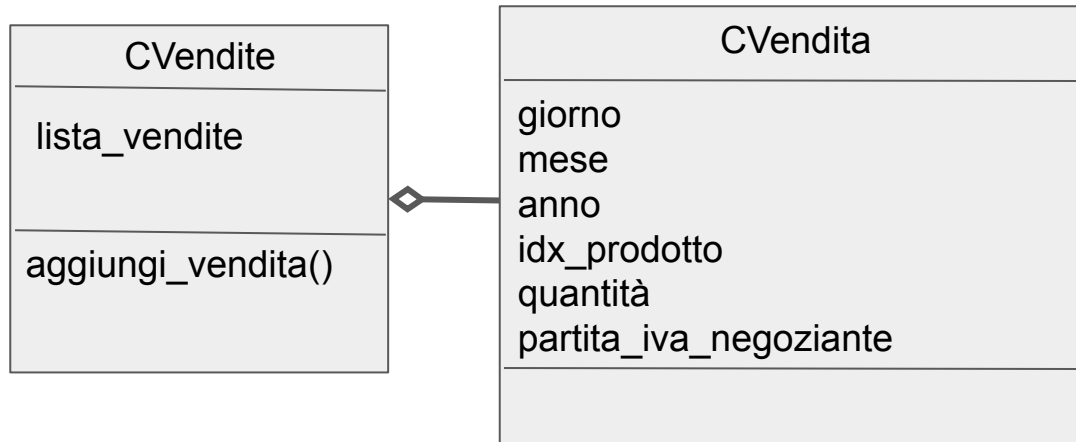
Per la vendita di ogni prodotto, da specifiche dobbiamo memorizzare le seguenti informazioni: giorno, mese anno, idx_prodotto, quantità, partita_iva_negoziante.

CVendite inoltre deve permettere di memorizzare più vendite.



La Programmazione ad Oggetti in Python

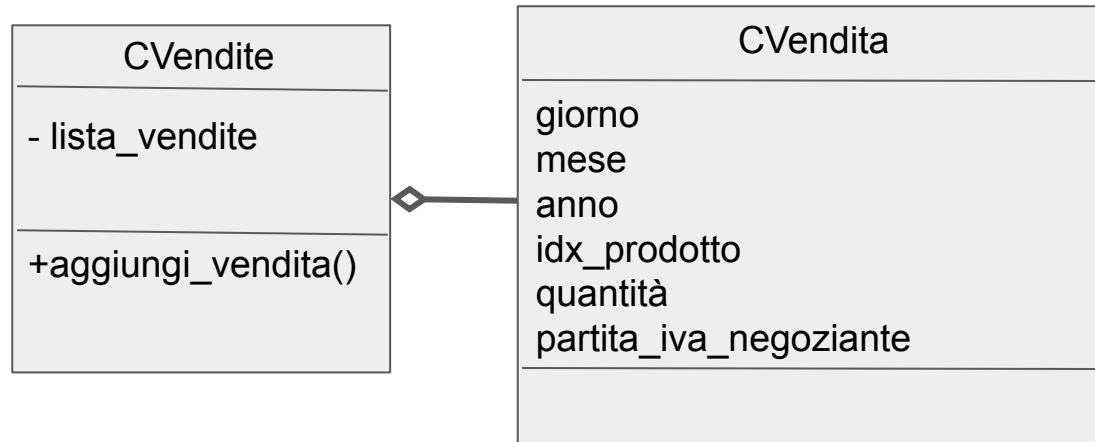
Attributi e metodi di CVendite devono essere pubblici o privati?



La Programmazione ad Oggetti in Python

Attributi e metodi di CVendite devono essere pubblici o privati?

Vogliamo fornire un metodo per inserire oggetti di tipo vendita quindi quello sarà pubblico, mentre la lista_vendite viene utilizzata solo dalla classe stessa, quindi sarà privata.



La Programmazione ad Oggetti in Python

Attributi e metodi di CVendita devono essere pubblici o privati?

Quali attributi ha senso che vengano modificati e quali è meglio vengano inseriti una volta e poi non vengano più modificati (di sola lettura)?

Per quali dobbiamo eseguire il controllo dell'input?

giorno

mese

anno

idx_prodotto

quantità

partita_iva_negoziante

La Programmazione ad Oggetti in Python

Attributi e metodi di CVendita devono essere pubblici o privati?

Quali attributi ha senso che vengano modificati e quali è meglio vengano inseriti una volta e poi non vengano più modificati?

Per quali dobbiamo eseguire il controllo dell'input?

Possiamo far sì che possano essere tutti modificabili, in modo che se l'impiegato li inserisce in modo errato possa modificarli.*

I loro valori devono però essere controllati, quindi possiamo definirli come privati e andremo poi a definire i metodi getter e setter.

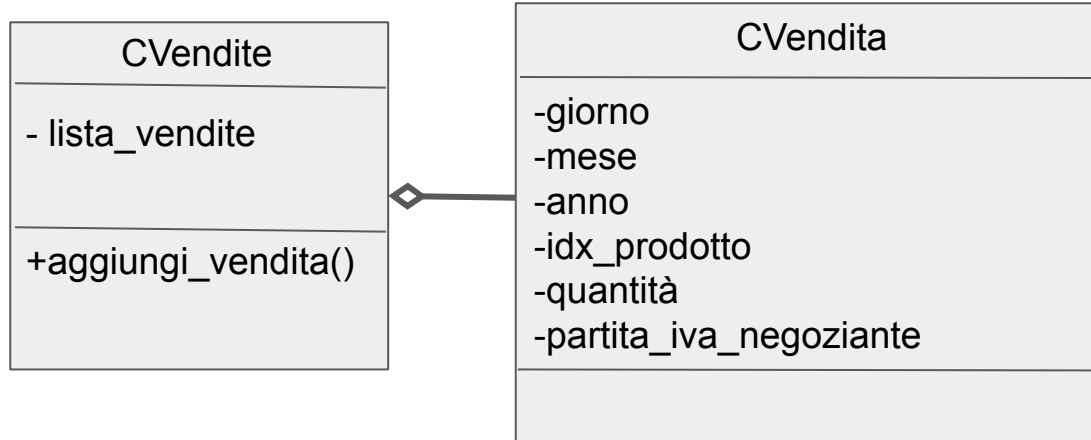
La Programmazione ad Oggetti in Python

*l'attributo più critico è l'idx_prodotto perchè compare sia nella classe CProdotto che in quella delle Vendita.

Però gli idx dei prodotti generalmente non vengono modificati.
Quindi non ha senso fornire all'utente dei metodi per modificare l'idx in modo coerente nelle due classi.

Se ha inserito un idx errato è quasi sicuramente dovuto ad un typo.

La Programmazione ad Oggetti in Python



La Programmazione ad Oggetti in Python

“Il prezzo di vendita dipende dal modello del prodotto, mentre il costo di produzione dipende dal tipo di prodotto:

*Tutti gli smartphone hanno un costo di produzione che dipende dal **costo dei componenti elettronici** + 500 euro di manodopera.*

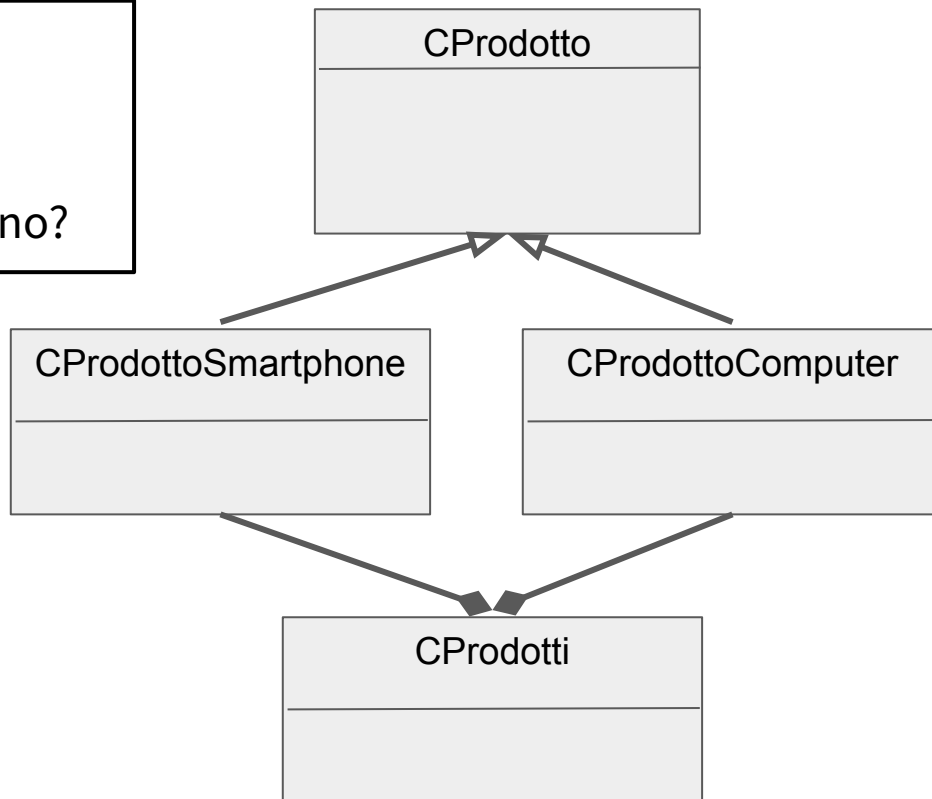
*Tutti i computer hanno un costo di produzione che dipende dal **costo dei componenti elettronici** + 20% del prezzo del prodotto. ”*

*“Per ora l’azienda vi chiede di considerare solo smartphone e pc come prodotti ma vi specificano che **in futuro hanno intenzione di** ampliare questo programma per **inserire tipi differenti di prodotti**, come le smart tv.”*

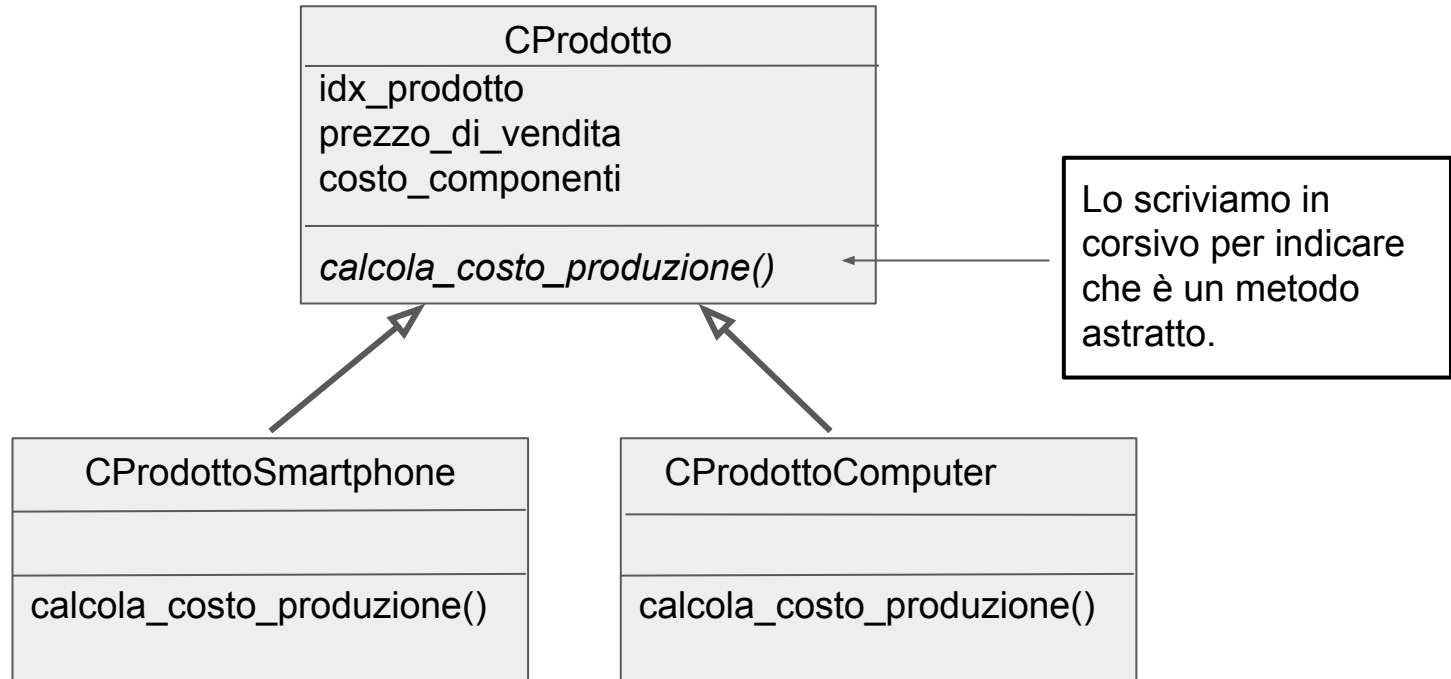
La Programmazione ad Oggetti in Python

Abbiamo definito questo diagramma.

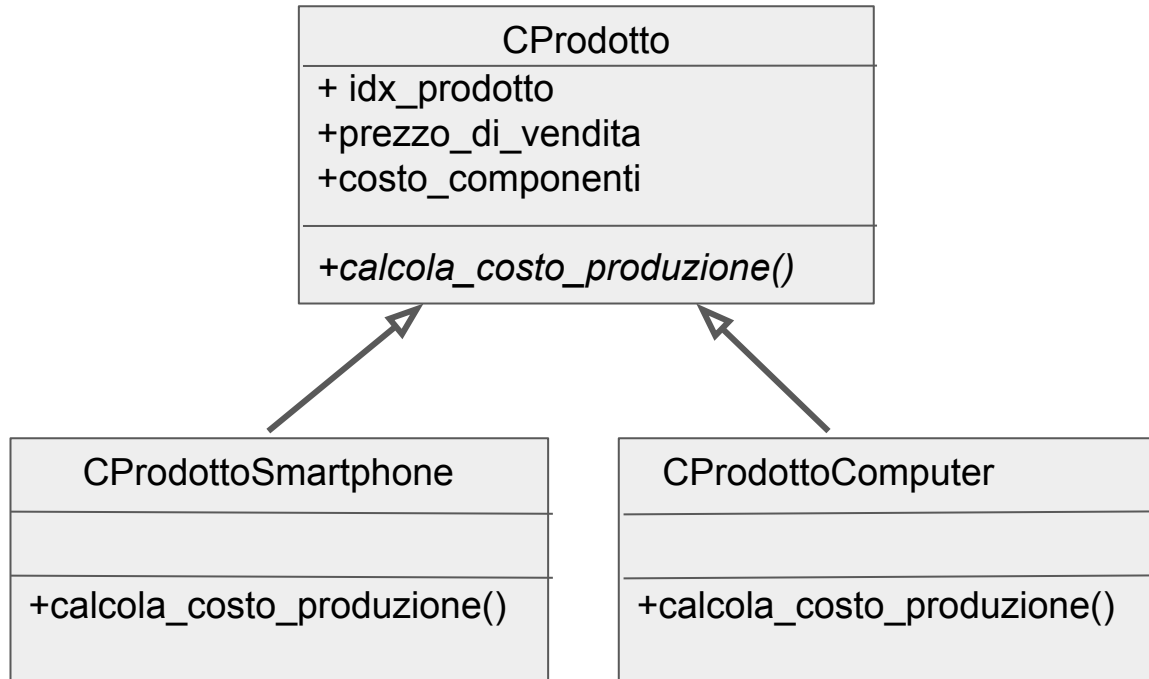
Quali metodi/attributi servono?



La Programmazione ad Oggetti in Python



La Programmazione ad Oggetti in Python



La Programmazione ad Oggetti in Python

CProdotti
-lista_prodotti
+ aggiungi_prodotto()

La Programmazione ad Oggetti in Python

“Deve inoltre **permettere di creare un grafico** dei ricavi ottenuti in un anno scelto. **Il ricavo per la vendita di un’unità di prodotto** è pari al **prezzo di vendita** meno il **costo di produzione**. ”

CCreatoreGrafici
+crea_grafico_ricavi_anno(anno, CVendite, CProdotti)

La Programmazione ad Oggetti in Python

Ci sono **altri metodi/attributi** che possono esserci **utili**? (Per rendere il programma più efficiente o per scomporre un metodo in sottoproblemi)?

La Programmazione ad Oggetti in Python

Ci sono **altri metodi/attributi** che possono esserci **utili**? (Per rendere il programma più efficiente o per scomporre un metodo in sottoproblemi)?

CCreatoreGrafici
+crea_grafico_ricavi_anno(anno, CVendite, CProdotti)

Creare un grafico dei ricavi di un anno è un'operazione complessa..
Per poter creare questo grafico dobbiamo ricavarci i dati...

La Programmazione ad Oggetti in Python

Dobbiamo crearci una lista che contiene i ricavi per ogni mese.

Per ogni vendita effettuata nell'anno scelto, dobbiamo:

- cercare l'oggetto prodotto con l'idx specificato nella vendita
- usarlo per calcolarci il ricavo dato da quella vendita e andare a sommarlo a quelli del mese opportuno.

Possiamo quindi creare due metodi privati utili:

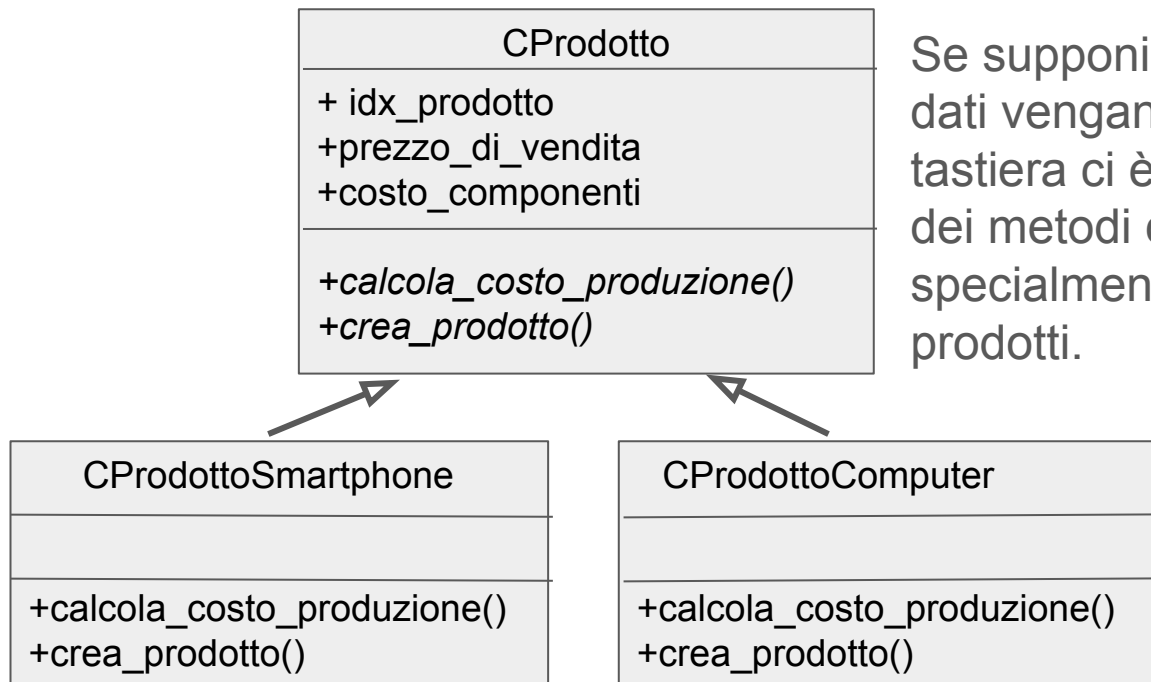
- calcola_ricavi_anno(anno, CVendite, CProdotti)
- calcola_ricavo_vendita(CVendita, CProdotto)

La Programmazione ad Oggetti in Python

CCreateoreGrafici
<ul style="list-style-type: none">+crea_grafico_ricavi_anno(anno, CVendite, CProdotti)-calcola_ricavi_anno(anno, CVendite, CProdotti)-calcola_ricavo_vendita(CVendita, CProdotto)

Nota: Dove è meglio si trovi `calcola_ricavo_vendita` dipende dalle assunzioni riguardo al problema. Questa soluzione assume che i ricavi delle vendite siano utili solo per la creazione dei grafici. Altrimenti i metodi relativi dovrebbero trovarsi nella classe `CVendita`.

La Programmazione ad Oggetti in Python



Se supponiamo che i dati vengano acquisiti da tastiera ci è utile avere dei metodi costruttori, specialmente per i prodotti.

La Programmazione ad Oggetti in Python

Può essere utile far sì che le istanze di alcune delle classi siano **iterabili**?

La Programmazione ad Oggetti in Python

Può essere utile far sì che le istanze di alcune delle classi siano **iterabili**?

-Per trovare le vendite dell'anno scelto dobbiamo scorrere la lista di vendite memorizzate in CVendite.

-Per trovare il prodotto con uno specifico idx e recuperarci il prezzo di vendita e il costo di produzione dobbiamo scorrere la lista di prodotti memorizzata in CProdotti.

E' quindi utile rendere CVendite e CProdotti iterabili.

La Programmazione ad Oggetti in Python

Può essere utile definire **oggetti di tipo eccezione**?

Ci sono delle operazioni che possono portare ad un'eccezione?

La Programmazione ad Oggetti in Python

Può essere utile definire **oggetti di tipo eccezione**?

Ci sono delle operazioni che possono portare ad un'eccezione?

Il metodo `calcola_ricavi_anno` deve:

Creare una lista che contiene i ricavi per ogni mese.

Per ogni vendita effettuata nell'anno scelto, dobbiamo:

- cercare l'oggetto prodotto con l'idx specificato nella vendita

- usarlo per calcolarci il ricavo dato da quella vendita e andare a sommarlo a quelli del mese opportuno.

La Programmazione ad Oggetti in Python

Se l'idx non viene trovato significa che c'è un problema nei dati che sono stati inseriti.

Possiamo quindi sollevare un'eccezione e può essere comodo creare una classe eccezione ad-hoc.