

ELENCO DEGLI ARGOMENTI TRATTATI

AMBIENTE DI SVILUPPO IDLE PYTHON

Finestre Shell e Editor.

Creare, memorizzare ed eseguire file Python (.py).

Attivare l'autocompletamento del codice e i suggerimenti contestuali.

Modalità di segnalazione di segnalazione degli errori lessicali e sintattici.

LE NOZIONI BASE DEL LINGUAGGIO PYTHON

Punti di forza e debolezza di Python rispetto a C++

Blocco principale (main) e indentazione.

Commenti.

Definizione di tipo di dato.

Esempi di dati appartenenti a tipi predefiniti in Python :

- intero(int), decimale(float), booleano(bool), stringa(str).

Definizione di funzione e argomento.

Chiamata(esecuzione) di una funzione

Funzioni built-in: funzioni predefinite in Python

- Di uso generale: input, print, len, type, id, len;
- Di conversione: chr, float, int, ord;
- Di calcolo: abs, pow, round.

Esercizi sull'uso di queste funzioni built-in.

Ripasso della definizione di variabile in C++

Python, esempi di assegnazione di espressioni:

- numeriche che utilizzano gli operatori aritmetici;
- di concatenazione di stringhe e dati numerici;
- booleane che utilizzano gli operatori logici;
- uso degli indici nelle stringhe
- assegnazioni multiple: scambio di variabili

Caratteristiche tipiche delle Variabili Python:

- Utilizzare la funzione id per identificare la porzione di memoria associata a un variabile.
- Immutabilità di una variabile di tipo predefinito. Esempi con le stinghe;
- Utilizzare la funzione type per la determinazione del tipo di una variabile;

Cenni alla programmazione a oggetti:

- In Python tutti i tipi sono delle classi: definiscono i dati come insieme di valori e di funzioni.
- Un dato Python viene chiamato oggetto e le sue funzioni si chiamano metodi
- Esercizi in cui si utilizzano metodi delle stringhe:

DAL PROBLEMA AL PROGRAMMA

Descrizione delle diverse classificazioni dei dati di un problema:

- Di input: dati necessari che vengono inseriti da chi utilizza il software
- Di output: dati che devono essere presentati a chi utilizza il software
- I vincoli: informazioni utili per l'elaborazione dei dati di input
- Le ipotesi aggiuntive: dati che non sono contenuti nel testo ma il cui valore viene determinato dal programmatore per presentare al meglio i dati di output

Esempi di strategie per risolvere un problema:

- Partire dalla codifica di algoritmi di problemi simili a quello da risolvere, Apportare successivamente modifiche per adeguare la codifica alle richieste del problema assegnato;
- Rappresentare il criterio risolutivo con un modello grafico, matematico, insiemistico.....;
- Arrivare a una soluzione parziale semplificando il set dei dati iniziali o riducendo il set dei dati di output. Successivamente si passa a modificare la codifica così ottenuta in modo da renderla più completa.

Utilizzare tabelle di traccia per l'individuazione degli errori logici di programmazione.

Esempi di problemi che richiedono Le strutture di controllo:

- I connettivi logici
- La selezione
- Le selezioni annidate
- La selezione multipla
- La ripetizione while
- La ripetizione con controllo della fine dell'input
- La ripetizione for in range
- Salti incondizionati: break e continue

LE STRUTTURE DATI

Esempi di problemi che richiedono la l'elaborazione di molti dati dello stesso tipo

Concetto di array monodimensionale e bidimensionale(matrice)

Le liste:

- Definizione dell'oggetto di classe list,
- Metodi dell'oggetto di classe list
- Le liste come array
- Ciclo per elaborare tutti gli elementi di un array; for in lista
- Funzioni di oggetto list che ne modificano il contenuto: append(), clear(), insert(), extend(), insert(), pop(), remove(), copy()
- Funzioni di oggetto list che restituiscono informazioni: index(), count()
- Funzioni di oggetto list che ordinano i dati contenuti: sort(), reverse()
- Conversione di liste in array e viceversa

Esempi di problemi che richiedono l'elaborazione:

- di strutture con dati di tipo diverso;
- di più array contemporaneamente;
- di array bidimensionali

LE FUNZIONI (SOTTOPROGRAMMI)

Definizione di libreria (modulo)

Esercitazioni pratiche sull'uso delle funzioni importate dalle seguenti librerie:

- math: exp, log, sqrt, trunc, fabs, ceil, floor
- random: randrange, randint, random

Lo sviluppo top-down

- suddividere il problema in sottoproblemi di cui si conosce il funzionamento

Creare una funzione:

- definire in nome del blocco della funzione
- indicare eventuali parametri
- utilizzo della parola chiave return
- visibilità delle variabili: variabili locali e variabili globali
- uso nella funzione dei parametri immutabili e di tipo list

Eseguire una funzione:

- passaggio degli argomenti
- restituzione del dato specificato nel return

Esempi di problemi il cui codice è strutturato in funzioni da richiamare nel main

Ricodificare problemi con un solo blocco main, mediante la metodologia top-down

FILE TESTO

Definizione di file testo e file logico

Oggetto file logico creato all'apertura del file fisico:

- Funzione open in modalità lettura
- Funzione open nelle diverse modalità di scrittura

Ciclo for per la lettura delle linee di testo contenute nel file

Metodi del file logico:

- Di lettura : fread(), freadline(),freadlines(),fread(n)
- Di scrittura: fwrite(),in base alla modalità di apertura
- Chiusura : close()

Funzioni del modulo os:

- Cancellazione : remove("file"), rmdir("directory")
- Ricercare; os.path.exists("file o directory")

INTRODUZIONE ALLA PROGRAMMAZIONE A OGGETTI IN PYTHON

Definizione di classe, attributi e metodi.

Classi Python per la definizione dei tipi di dato: int,str,bool,float

Definizione di oggetto come istanza di una classe

Definizione di variabile in Python: riferimento a un oggetto memorizzato in memoria

Import di librerie di classi

Definizione di oggetto

Utilizzo di classi predefinite in Python

Creazione di oggetti con classi definite

CLASSI DEFINITE DALL'UTENTE PROGRAMMATORE

Fasi della programmazione a oggetti:

- Individuare uno o più oggetti reali definiti nel testo del problema assegnato
- Esempi di oggetti fisici e di oggetti astratti
- Rappresentare le caratteristiche dell'oggetto con il diagramma della classe: attributi e metodi
- Codificare i metodi __init__ e __str__ e tutti quelli presenti nel diagramma della classe
- Definire metodi e attributi privati della classe
- Definire metodi globali della classe

Ereditarietà:

- Derivazione di una classe da una o più classi dette superclassi
- Super: riferimento alla superclasse
- Chiamata al metodo costruttore di una superclasse

Modello per rappresentare la gerarchia tra più classi

Vantaggi della dell'ereditarietà

Esempi di problemi che prevedono l'elaborazione di oggetti:

- della stessa classe
- di due o più classi diverse
- di una classe derivata
- di più classi derivate dallo stesso superclasse
- di una classe derivata da più superclassi

L'INTERFACCIA GRAFICA(GUI) CON IL MODULO TKINTER

Componenti grafici: definizione di widget e di finestra dell'applicazione.

Modalità per importare da tkinter le classi dei componenti grafici.

Metodi e attributi delle seguenti Classi grafiche:

- widget di output/input: Label e Entry
- widget per la gestione degli eventi: Button
- widget dove inserire i componenti di interazione: Frame
- Finestra dell'applicazione dove inserire un Frame: TK

Modalità di progettazione di semplici Gui

Codifica della classe Finestra necessaria per creare l'oggetto GUI progettato

Realizzazione di applicazioni ludiche:

- Ottimizzazione dei widget all'interno della GUI
- Gestione degli eventi in base alla fase di gioco
- Gestione di elementi che valutano le abilità del giocatore: punteggi, livelli di gioco;
- Condizioni per la fine del gioco.

Gli argomenti trattati e gli esempi svolti si trovano nel libro di testo e nella piattaforma online W3schols.

I problemi da codificare sono stati presi in buona parte dal libro di testo

Catania 12/06/2024

I proff.
Salvatore Antonio arcidiacono
Giuseppe Biuso