﻿Giorno 3

le entita astratte le modello con le interfacce.

implements: questa classe è capace di fare cose che sono presenti nell interfaccia.

Voglio vedere poca ereditarietà e molte interfacce

in java l ereditarieta è singola cieo nn posso ereditare da piu di una classe ma posso implementare tutte le interfacce che voglio

qualità interna : miglioro il codice senza cambiare risultato

qualita esterna : miglioro il codice , cambiamenti visibili anche dall utente che usa l app

Gli oggetti,a seconda della classe, possono esserere:

mutabili:ha uno stato che puo variare nel corso del tempo

immutabili:non cambia lo stato interno(poiche nn ci sono metodi che cambiano lo stato )

Le stringhe in java sono oggetti immutabili

è la reference che è mutabile ma non l oggetto stringa

-->Final serve per fare le reference immutabili e non gli oggetti

StingBuilder è una stringa mutabile

String è una stringa immutabile

in java tutte le variabili sono passate per copia

un metodo statico è una procedura

regular expression :matching capacita di andare a dire se ce una corrispondenza con un testo

parola

.\* trova tutto dopo la parola

.el : tutto le parole in cui trovo el con un carattere prima

costruttore privato: posso usare new solo nella mia classe in un costrutttore private

ARRAY

in java è un oggetto

la dimensione è un attributo che interessa il tipo dinamico e non il tipo statico

sono un aggregazione vettoriale solo di tipi primitivi oppure reference

l array è mutabile

è immutabile per quanto riguarda la sua dimensione

LIST

tutte le collezioni sono in java.util

tutte le collezioni sono definite da interfacce

la lista è mutabile

posso inserire qualsiasi oggetto apparentemente(lista eterogenea)

in java ce una classe object che è radice di tutti gli oggetti

le

classi wrapper sono tutte immutabili

un info del tipo generico la tiene solo il compilatore non vanno nel bytecode

la covarianza ce solo con gli array ma non sulle liste

gli array sono covarianti

i tipi generici nn finisco a run time

lista:

è una collezione di ogge

è dinamica

è una collezione che accetta duplicati (posso inserire due stringhe uguali)

mantiene l ordine costituito

linked list:

accesso casuale

differenza di performance con la list

devo scorrere sempre dall inizio qnd cerco un elemento

vantaggio se dobbiamo fare add o remove frequenti nn alla fine delle posizioni

stessa interfaccia di list

SET LIST

non sopporta l ordine ...cioe nn posso parlare di terzo elemento

l unico problema che mi pongo è se contiene o no un certo oggetto

non accetta duplicati ...se inserisco due elementi uguali ne tiene conto solo di uno

non è deterministica

non rispetto l ordine

utilizza eguaglianza semantica:

TREESET

rispetta sempre l ordine naturale..nel caso di stringhe ordine alfabetico

si puo utilizzare solo con oggetti per cui è definito un ordine naturale...ad esempio nn posso usarlo con le Car.

MAP

è una collezione di coppie

si agg con .put(kiave ,valore)

non è garantito nessun ordine

esiste una treemap che mantiene l ordine naturale

OVERRIDE

ridefinire metodi

hashMap

chiavi ed elementi non ammettono duplicati

si usa solo con oggetti immutabili

ALGORITMO DI HASHING

due istanze di ogg che sono equals devono produrre ssempre lo stesso hashcode

due ogg nn equals farebbero meglio a dare una distribuzine di num interi piu sparpagliata possibile per riempire i bucket uniformemente

tutti gli ogg java utilizzati come chiave in hashmap o implementazioni hash bisogna implementare il metodo hashcode

sulle collection come set e map dove ci sono vincoli di unicità per vedere se due ogg sono uguali ce il metodo equals oppure ce il metodo hascode che calcola un codice univoco per ogni ogg che viene usato per migliorare il metodo di ricerca xo è molto fragile

COSTRUTTO LAMBDA

modo per implementare al volo un interfaccia composta da un unico metodo