Cuori solitari (versione 0.0)

- Si richiede lo sviluppo di un programma che verifica la possibile affinità tra due “cuori solitari” in base ad alcuni dati personali
- Nella prima versione si suppone che i dati richiesti a ciascuno siano:
  - uno pseudonimo
  - il sesso
  - l’età

Cuori solitari (versione 0.0)

- Il programma, dopo aver presentato un saluto iniziale, richiede i dati di due soggetti e presenta a video un messaggio del tipo: “XXX (non) è potenzialmente affine con YYY” dove XXX e YYY sono gli pseudonimi dei due soggetti
- Nella versione 0.0 due soggetti sono affini se:
  - sono di sesso diverso
  - la differenza di età è minore di una certa soglia (stabilita come costante a discrezione del programmatore)

Suggerimenti progetto (primo passo)

- Il primo passo consiste nell'identificazione delle classi necessarie
- E' possibile individuare:
  - una classe (istanziabile) per rappresentare ciascun "cuore solitario"
  - una classe (tutta static) per il main e per altri metodi di utilità specifici del problema (p.e. il saluto, la creazione di un cuore solitario)
  - una classe (tutta static) per i metodi di utilità generale, quali le letture dei dati in input

Suggerimenti progetto (secondo passo)

- Progettare un programma non significa solo pensare a come sarà fatto ma anche a come sarà testato
- E' opportuno definire un piccolo piano di test:
  - una prova in cui si inseriscono i dati di due persone tra le quali c'è affinità
  - una prova in cui si inseriscono i dati di due persone tra le quali non c'è affinità perché sono dello stesso sesso
  - una prova in cui si inseriscono i dati di due persone tra le quali non c'è affinità perché la differenza di età è troppa
  - tre prove nelle quali si inseriscono i dati delle stesse persone di cui sopra ma in ordine inverso (in ciascun caso il risultato non deve cambiare)

Suggerimenti progetto (terzo passo)

- Una possibile realizzazione della classe che rappresenta ciascun cuore solitario comprende:
  - tre attributi non static + qualche costante
  - un costruttore per inizializzare gli attributi
  - un metodo che riceve in ingresso il riferimento ad un altro cuore solitario e restituisce un valore boolean corrispondente alla possibile affinità o meno
  - un metodo che restituisce la rappresentazione come String (in questo caso, per motivi di privacy, si restituisce solo lo pseudonimo)

Scelte “obbligate” e scelte libere

- Il tipo dell’attributo che rappresenta lo pseudonimo è “naturalmente” String
- Il tipo dell’attributo che rappresenta l’età è “naturalmente” int (ma basta anche short o byte)
- La scelta è decisamente più libera sul terzo attributo: un char ('M' o 'F'), una String ("M" o "F"), un int (p.e. 1 per M, 2 per F), un enum definito appositamente, ...
- Nei primi tre casi dovranno essere definite delle costanti opportune

Suggerimenti progetto (quarto passo)

- Una possibile realizzazione del metodo main prevede che esso:
  - presenti un saluto
  - effettui la richiesta dati e creazione della prima istanza di cuore solitario
  - effettui la richiesta dati e creazione della seconda istanza di cuore solitario
  - verifichi l'affinità tra i due e presenti un messaggio a video di conseguenza

Suggerimenti progetto (quinto passo)

- Pertanto la classe che contiene il main, potrà contenere anche:
  - un metodo per il saluto iniziale
  - un metodo per la richiesta dati e creazione di un'istanza di cuore solitario (da usare due volte)
  - eventualmente (ma non è indispensabile) anche la parte di verifica e stampa dell'affinità può essere trattata in un metodo a parte

Suggerimenti progetto (sesto passo)

- La classe con i metodi di utilità generale potrebbe contenere:
  - un metodo per acquisire da tastiera una String
  - un metodo per acquisire da tastiera un numero intero
  - eventualmente un metodo per acquisire da tastiera un singolo char

Sviluppo graduale

- E’ consigliabile procedere per gradi nello sviluppo
- Per esempio si può fare una prima versione nella quale la classe istanziabile ha il solo attributo pseudonimo, lo si acquisisce da tastiera (per le due istanze) e lo si stampa
- Quindi una versione in cui ci sono tutti gli attributi e vengono acquisiti (e magari stampati) senza fare ancora la verifica di affinità
- Infine si aggiungerà la verifica di affinità e si faranno le prove complete previste dal piano di test

Cuori solitari (versione 0.1)

- Se la versione 0.0 è stata completata con successo, possiamo pensare a migliorarla in varie direzioni
- Una prima direzione è quella di aumentare i controlli sui dati inseriti dall'utente:
  - supponiamo che il servizio sia riservato a maggiorenni (quindi l'età deve essere almeno 18) e rifiutiamo valori di età implausibili perché troppo alti (p.e. superiori a 120)
  - potremmo controllare che lo pseudonimo non sia una stringa vuota o composta da soli caratteri spazio e tab (viene utile il metodo trim() della classe String)
  - se richiediamo un solo carattere per rappresentare il sesso verifichiamo che sia esattamente M o F (eventualmente anche minuscoli), se abbiamo scelto un'altra rappresentazione, servirà un controllo analogo

Cuori solitari (versione 0.1)

- La versione 0.1 richiede sostanziali aggiunte essenzialmente alla classe contenente i metodi di acquisizione dati da tastiera: bisogna crearne nuove versioni capaci di ricevere più argomenti e fare più controlli
- Anche la classe con il main dovrà essere adattata per invocare le nuove versioni dei metodi di acquisizione, passando loro i parametri aggiuntivi
- La classe che rappresenta i cuori solitari non dovrebbe cambiare per nulla

Sviluppo graduale 0.1

- Ovviamente si suggerisce di aggiungere un nuovo controllo per volta e per ciascuno definire dei casi di prova in cui l'utente "sbaglia" e dei casi di prova in cui l'utente si comporta correttamente
- Chi va piano . . . va veloce

Cuori solitari (versione 0.2)

- Un’altra possibile direzione di miglioramento prevede di aggiungere nuovi attributi nel profilo personale dei quali tenere conto nella verifica di affinità
- Ad esempio si potrebbe richiedere per ciascun cuore solitario:
  - il suo segno zodiacale di nascita
  - il segno zodiacale del/della partner ideale

Cuori solitari (versione 0.2)

- La versione con i segni zodiacali richiede:
  - di decidere come rappresentare i segni stessi (ci sono varie alternative possibili, viene naturale pensare ad enum)
  - di aggiungere due attributi nella classe istanziabile
  - di modificare di conseguenza il costruttore e il metodo di acquisizione dati presente nel main
  - di modificare il metodo di verifica di affinità aggiungendo il controllo sul reciproco “gradimento zodiacale”
  - di definire un nuovo piano di test specifico per i segni zodiacali

Sviluppo graduale 0.2

C’e’ bisogno di dirlo?