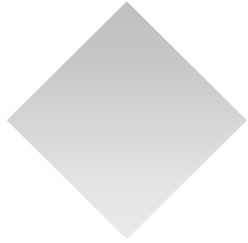


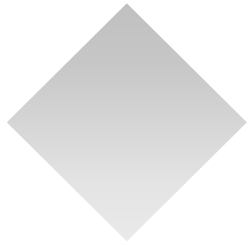
# Tamagotchi

- Si richiede lo sviluppo di un programma che permette all'utente di interagire con un Tamagotchi (in futuro magari più di uno)
- Un Tamagotchi è un'entità software in grado di recepire stimoli dall'esterno che determinano la sua sopravvivenza e il suo grado di benessere
- Per semplicità si suppone che:
  - gli stimoli ricevuti possano essere carezze o biscotti
  - il grado di benessere sia rappresentabile tramite due soli valori numerici (grado di soddisfazione affettiva, grado di sazietà)



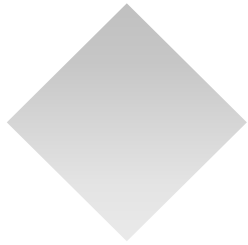
# Vita del Tamagotchi

- Si suppone che gli stimoli esterni siano forniti come numeri interi (numero di carezze o numero di biscotti)
- I due valori interni del Tamagotchi variano in base agli stimoli come segue:
  - Se riceve carezze, aumenta il grado di soddisfazione affettiva in base al numero di carezze ricevute ma diminuisce il grado di sazietà (p.e. della metà delle carezze ricevute)
  - Se riceve biscotti, aumenta il grado di sazietà secondo la regola che ogni biscotto mangiato aumenta del 10% il grado di sazietà precedente. Diminuisce però il grado di soddisfazione affettiva (p.e. di un quarto dei biscotti ricevuti)



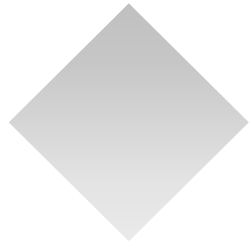
# Vita del Tamagotchi

- I due valori interni variano tra 0 e un valore massimo (p.e. 100) oltre il quale ulteriori stimoli non hanno effetti
- Il Tamagotchi muore quando:
  - uno dei due valori interni raggiunge 0
  - il grado di sazietà raggiunge il massimo (troppo cibo)
- Il Tamagotchi è infelice quando:
  - il grado di soddisfazione è sotto una certa soglia (p.e.  $<30$ )
  - il grado di sazietà è troppo basso o troppo alto (p.e.  $<30$  o  $>90$ )



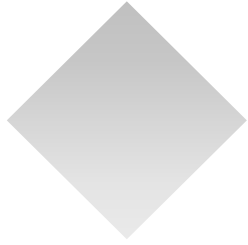
# Tamagotchi Main

- Il programma, dopo aver presentato un saluto iniziale, richiede di creare un Tamagotchi chiedendo all'utente i valori iniziali dei gradi di soddisfazione e di sazietà e un nome (se vogliamo battezzarlo)
- Segue un ciclo nel quale l'utente deve scegliere fra tre alternative:
  - dare carezze
  - dare biscotti
  - uscire dal programma



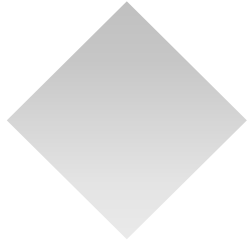
# Tamagotchi Main

- In base alla scelta dell'utente viene estratto un numero casuale di biscotti o carezze da somministrare al Tamagotchi, variando il suo stato di conseguenza
- Se a seguito della variazione il Tamagotchi muore, il programma termina producendo un opportuno messaggio a video
- Altrimenti il ciclo prosegue e l'utente viene informato se il Tamagotchi è infelice



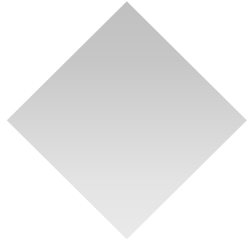
# Suggerimenti progetto (primo passo)

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



# Suggerimenti progetto (secondo passo)

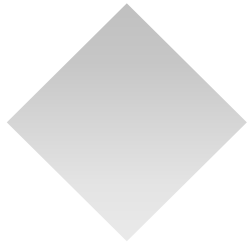
- Una possibile realizzazione della classe che rappresenta un Tamagotchi comprende:
  - tre attributi non static + qualche costante
  - un costruttore per inizializzare gli attributi
  - uno o più metodi void per ricevere gli stimoli e aggiornare lo stato interno (p.e. uno per biscotti, uno per carezze)
  - uno o più metodi per verificare la sopravvivenza e l'infelicità del Tamagotchi (p.e. due metodi boolean per le due condizioni)
  - un metodo che restituisce la rappresentazione come String



## Suggerimenti progetto (terzo passo)

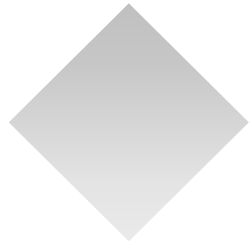
- Una possibile realizzazione del metodo main prevede che esso:
  - presenti un saluto
  - effettui la richiesta dati e creazione del Tamagotchi
  - svolga il ciclo di interazione con l'utente, controllando le condizioni di uscita
  - verifichi lo stato del Tamagotchi a ogni passo e riferisca a video





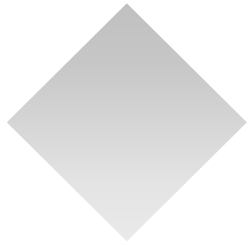
# Sviluppo graduale

- E' consigliabile procedere per gradi nello sviluppo
- Per esempio si può fare una prima versione nella quale il Tamagotchi non muore mai e non è mai infelice ma modifica solo il suo stato interno
- Quindi una versione in cui c'è solo il metodo di verifica di sopravvivenza
- Quindi una versione in cui c'è anche il metodo di verifica di infelicità
- Infine si faranno le prove complete previste da un piano di test



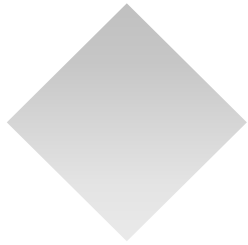
# Il piano di test

- Il piano di test è reso più complicato dalla presenza di un'estrazione casuale di valori. Volendo si può sostituire temporaneamente l'estrazione casuale con un valore inserito da tastiera per avere dei casi di test noti
- Anche con l'estrazione casuale è possibile verificare almeno il corretto comportamento in situazioni estreme:
  - se gli do solo carezze, prima o poi muore di fame
  - se gli do solo biscotti, prima o poi muore di tristezza o di troppo cibo
- Stampando a video i valori interni (anche se non richiesto dal testo) si può comunque verificare se il comportamento passo-passo è come previsto



# Tamagotchi: più controlli

- Se la versione base è stata completata con successo, possiamo pensare a migliorarla in varie direzioni
- Una prima direzione è quella di effettuare i controlli sui dati inseriti dall'utente (se non l'abbiamo ancora fatto) in modo del tutto analogo a quanto visto nella precedente esercitazione
- L'uso di una classe già predisposta potrebbe rendere praticamente nullo fin dall'inizio lo sforzo per effettuare questi controlli



# Tamagotchi: evoluzione

- Un'altra possibile direzione di estensione prevede di avere meccanismi di evoluzione del Tamagotchi più sofisticati
- Ad esempio l'effetto di carezze e cibo può dipendere in modo più complicato dal precedente stato del Tamagotchi: p.e. se ha già ricevuto molte carezze le successive hanno meno effetto
- In modo simile, anziché stabilire solo se è infelice o no si può definire un grado complessivo di benessere ottenuto come media pesata dei due gradi definiti in precedenza