### Una clase para probar

Probar una clase es similar a probar una función, porque gran parte del trabajo consiste en probar el comportamiento de los métodos de la clase. Sin embargo, hay algunas diferencias, así que vamos a escribir una clase para probar. Considera una clase que ayude a administrar encuestas anónimas:

**survey.py**

class AnonymousSurvey:  
 """Collect anonymous answers to a survey question."""  
  
❶ def \_\_init\_\_(self, question):  
 """Store a question, and prepare to store responses."""  
 self.question = question  
 self.responses = []  
  
❷ def show\_question(self):  
 """Show the survey question."""  
 print(self.question)  
  
❸ def store\_response(self, new\_response):  
 """Store a single response to the survey."""  
 self.responses.append(new\_response)  
  
❹ def show\_results(self):  
 """Show all the responses that have been given."""  
 print("Survey results:")  
 for response in self.responses:  
 print(f"- {response}")

Esta clase comienza con una pregunta de encuesta que tú proporcionas ❶ e incluye una lista vacía para almacenar las respuestas. La clase tiene métodos para imprimir la pregunta de la encuesta ❷, añadir una nueva respuesta a la lista de respuestas ❸, e imprimir todas las respuestas almacenadas en la lista ❹. Para crear una instancia a partir de esta clase, todo lo que tienes que proporcionar es una pregunta. Una vez que tengas una instancia que represente una encuesta concreta, muestra la pregunta de la encuesta con show\_question(), almacena una respuesta utilizando store\_response(), y muestra los resultados con show\_results().

Para demostrar que la clase AnonymousSurvey funciona, vamos a escribir un programa que utilice la clase:

**language\_survey.py**

from survey import AnonymousSurvey  
  
# Define a question, and make a survey.  
question = "What language did you first learn to speak?"  
language\_survey = AnonymousSurvey(question)  
  
# Show the question, and store responses to the question.  
language\_survey.show\_question()  
print("Enter 'q' at any time to quit.\n")  
while True:  
 response = input("Language: ")  
 if response == 'q':  
 break  
 language\_survey.store\_response(response)  
  
# Show the survey results.  
print("\nThank you to everyone who participated in the survey!")  
language\_survey.show\_results()

Este programa define una pregunta ("What language did you first learn to speak?") y crea un objeto AnonymousSurvey con esa pregunta. El programa llama a show\_question() para mostrar la pregunta y luego pide respuestas. Cada respuesta se almacena a medida que se recibe. Cuando se han introducido todas las respuestas (el usuario introduce q para salir), show\_results() imprime los resultados de la encuesta:

What language did you first learn to speak?  
Enter 'q' at any time to quit.  
  
Language: English  
Language: Spanish  
Language: English  
Language: Mandarin  
Language: q  
  
Thank you to everyone who participated in the survey!  
Survey results:  
- English  
- Spanish  
- English  
- Mandarin

Esta clase funciona para una simple encuesta anónima, pero supongamos que queremos mejorar AnonymousSurvey y el módulo en el que se encuentra, survey. Podríamos permitir que cada usuario introdujera más de una respuesta, podríamos escribir un método para listar sólo las respuestas únicas e informar de cuántas veces se ha dado cada respuesta, o incluso podríamos escribir otra clase para gestionar las encuestas no anónimas.

Implementar tales cambios podría afectar al comportamiento actual de la clase AnonymousSurvey. Por ejemplo, es posible que, al intentar permitir que cada usuario introduzca varias respuestas, cambiemos accidentalmente el modo en que se gestionan las respuestas únicas. Para asegurarnos de que no rompemos el comportamiento existente mientras desarrollamos este módulo, podemos escribir pruebas para la clase.

[anterior](c11_17.html)[Subtema 18 de 22: (Ver todo)](c11.html)[siguiente](c11_19.html)