

- `void guess(int number)`
 - informa a la clase del número que ha elegido el usuario
 - este número será informado a cada uno de los `SecretNumbers` que hay en el array extensible
 - para que siempre existan en el array el mismo número de números por descubrir, se **añadirán al array** tantos `SecretNumbers` como **números se hayan adivinado** correctamente en el paso anterior.
 - tened en cuenta que solamente se considerará un número como adivinado, a efectos de añadir nuevos números, la **primera vez** que esto ocurra, es decir, los números que ya habían sido adivinados en invocaciones anteriores, ya no contarán para calcular el número de elementos a añadir.
 - en caso de que **no quede espacio** para los nuevos `SecretNumbers`, la política de **redimensionado** será la de **doblar** el tamaño actual
 - **NOTA:** fijaos en que al añadir estos nuevos números secretos podemos tener que alguno de los nuevos números secretos a adivinar coincida con uno que ya habíamos adivinado previamente.
- `int numGuessedOk()`
 - devuelve el número de números secretos que se han adivinado correctamente.

Apartado 3: El programa principal (3 puntos)

El programa principal pedirá al usuario una línea en la que, separados por espacios, se entrarán los números a comprobar. El número inicial de secretos será 10. El programa, antes de finalizar, escribirá la cantidad de números adivinados correctamente.

Como siempre, podéis suponer que el formato de las entradas es el correcto.

```
public class GuessProgram extends CommandLineProgram {

    private static final int INITIAL_SIZE = 10;
    private Game game;

    public void run() {
        this.game = new Game(INITIAL_SIZE);
        ???
    }

    ???
}
```