

Aviso:

El objetivo de los siguientes ejercicios y cuestiones no es aprender a resolver mecánicamente preguntas de examen, sino comprender los contenidos del temario y aprender a realizar razonamientos idealmente por encima de las expectativas del profesor. De esta forma será menos el temario que deberás memorizar, y toda la asignatura parecerá conexas.

Por ello, algunas preguntas pueden parecer **complejas y con trampa**. Recuerda que no debes desanimarte, el objetivo final no es que las respondas correctamente (porque yo no te voy a evaluar), sino que intentes encontrar la respuesta y te topes con todas las dudas que pudieran surgirti durante el examen. **En clase resolveremos todos los que queráis.**

1. Describe con tus palabras en qué consiste la abstracción procedimental.
2. Describe con tus palabras en qué consiste la abstracción de datos.
3. Describe con tus palabras en qué consiste la abstracción de iteración.
4. Describe con tus palabras en qué consiste la abstracción por parametrización.
5. Describe con tus palabras en qué consiste la abstracción por especificación.
6. Describe con tus palabras la relación entre especificación, representación e implementación.
7. Describe con tus palabras qué es un TDA.
8. Queremos diseñar un tipo de datos abstracto que encapsule la siguiente funcionalidad:
Una lotería. Al resetear o inicializar el juego se introducen 50 bolas numeradas del 1 hasta el 50 en un bombo. Cada vez que se extrae una bola, se obtiene aleatoriamente una de las que aún quedan dentro del bombo. En cualquier momento podremos consultar si una bola con un número determinado ha salido o no. Si se han obtenido todas las bolas, no se pueden intentar extraer más.
 - a) Localiza las funciones mínimas que debe proporcionar este TDA.
 - b) Especifica completamente (el TDA en global y cada una de sus abstracciones procedimentales o procedimientos en particular) e independientemente de la representación e implementación este TDA.
 - c) Propón una representación (los datos que debería tener el TDA) que permitiera resolver este problema, indicando para qué y cómo se usarían.
 - d) Propón otra representación válida, si se te ocurre.
 - e) Se te haya ocurrido o no: ¿Sería necesario cambiar la especificación del TDA (apartado b) según la representación escogida (apartados c y d)?
9. Queremos diseñar un tipo de datos abstracto que encapsule la siguiente funcionalidad:
Un número entero de tamaño arbitrario hasta 100 dígitos.
 - a) Localiza las funciones mínimas que debe proporcionar este TDA.
 - b) Especifica completamente (el TDA en global y cada una de sus abstracciones procedimentales o procedimientos en particular) e independientemente de la representación e implementación este TDA.
 - c) Propón una representación (los datos que debería tener el TDA) que permitiera resolver este problema, indicando para qué y cómo se usarían.
 - d) Propón otra representación válida, si se te ocurre.
 - e) Se te haya ocurrido o no: ¿Sería necesario cambiar la especificación del TDA (apartado b) según la representación escogida (apartados c y d)?

10. a) Ordena de menor a mayor abstracción las siguientes especificaciones de funciones:

```
1.
/**
 * @brief Rellena un vector con 5 ceros.
 * @param v Vector a rellenar.
 * @pre El vector está inicializado.
 * @pre El vector tiene longitud al menos 5.
 * @post Las cinco primeras posiciones del vector valen ahora 0.
 *
 * Rellena el vector v con 5 ceros, en las posiciones 0 a 4 inclusive.
 */
void rellena_vector_5_ceros(int v[]);

2.
/**
 * @brief Rellena un vector de cualquier longitud con ceros.
 * @param v Vector a rellenar.
 * @param n Longitud del vector.
 * @pre El vector está inicializado.
 * @pre El vector tiene longitud mayor o igual a n.
 * @post Las n primeras posiciones del vector valen ahora 0.
 *
 * Rellena el vector v con ceros, en las posiciones 0 a n-1 inclusive.
 */
void rellena_vector_ceros(int v[],int n);

3.
/**
 * @brief Rellena el vector global vect con 5 ceros.
 * @pre El vector global vect tiene longitud al menos 5.
 * @post Las cinco primeras posiciones del vector global vect valen ahora 0.
 *
 * Rellena el vector global vect con 5 ceros, en las posiciones 0 a 4 inclusive.
 */
void rellena_vect_5_ceros();

2.
/**
 * @brief Rellena un vector de cualquier longitud con ceros.
 * @param v Vector a rellenar.
 * @param n Longitud del vector.
 * @param val Valor con el que se desea rellenar el vector.
 * @pre El vector está inicializado.
 * @pre El vector tiene longitud mayor o igual a n.
 * @post Las n primeras posiciones del vector valen ahora val.
 *
 * Rellena el vector v con val, en las posiciones 0 a n-1 inclusive.
 */
void rellena_vector(int v[],int n,int val);
```

b) ¿De qué tipo de abstracción estamos hablando?

c) ¿Cuál podemos aplicar en más casos?