

1. When is the best time to use a design pattern? Choose **two** answers.

1 / 1 балл

- ☐ For a problem that is unique to your program.
- ☒ For a commonly-encountered issue.
- ✔ Правильно

Correct! Design patterns will help you with issues that developers have encountered often. Remember that sometimes they need some adapting.
- ☐ When fixing spaghetti code
- ☒ When explaining a solution to your fellow developers
- ✔ Правильно

Correct! Design patterns are coding solutions!

2. What is the purpose of the Singleton pattern? Select the **two correct** answers.

1 / 1 балл

- ☐ to enforce collaboration of a class with only one other class
- ☐ to provide simple classes with only one method
- ☒ to enforce instantiation of only one object of a class
- ✔ Правильно

Correct. The Singleton pattern enforces one and only one instantiation of the Singleton class.
- ☒
- to provide global access to an object
- ✔ Правильно

Correct. The Singleton pattern makes the one instance of the Singleton class globally accessible.

3. What does it mean to "let the subclass decide" in the Factory Method Pattern?

1 / 1 балл

- ☐ the subclass will pass a parameter into a factory that determines which object is instantiated.
- ☒ the subclass defines the methods for concrete instantiation. As such, the type of object is determined by which subclass is instantiated.
- ☐ the subclass decides which object to create, but calls a method that is defined in the superclass to instantiate the class
- ✔ Правильно

Correct! This is how the subclass "decides." By selecting a subclass you are limited to its concrete instantiation method.

4. What do we call the creation of an object, for example, with the 'new' operator in Java?

1 / 1 балл

- ☒ concrete instantiation.
- ☐ manifestation
- ☐ object realization
- ☐ class creation
- ✔ Правильно

Correct! Instantiation is the act of creating an instance of a class, while concrete means the actual act of doing it (rather than speaking about it in general terms, like some interface for creating objects).

5. What are the advantages of the **Facade** pattern? Select the **three correct** answers.

1 / 1 балл

- ☒ The client and the subsystem are more loosely coupled
- ✔ Правильно

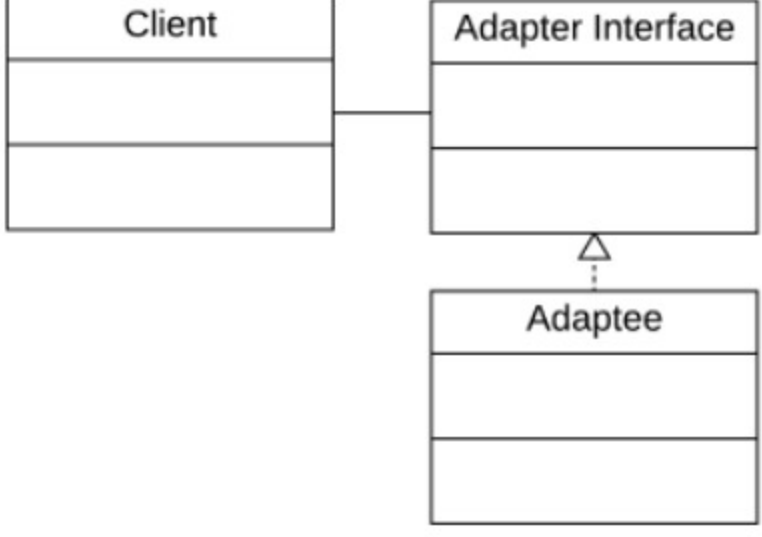
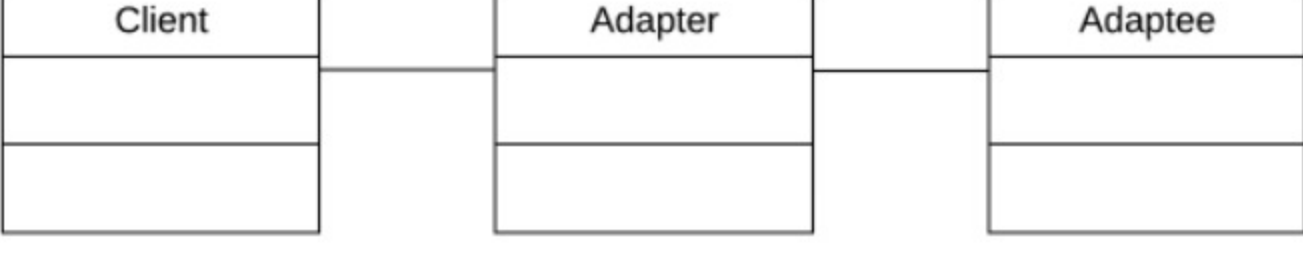
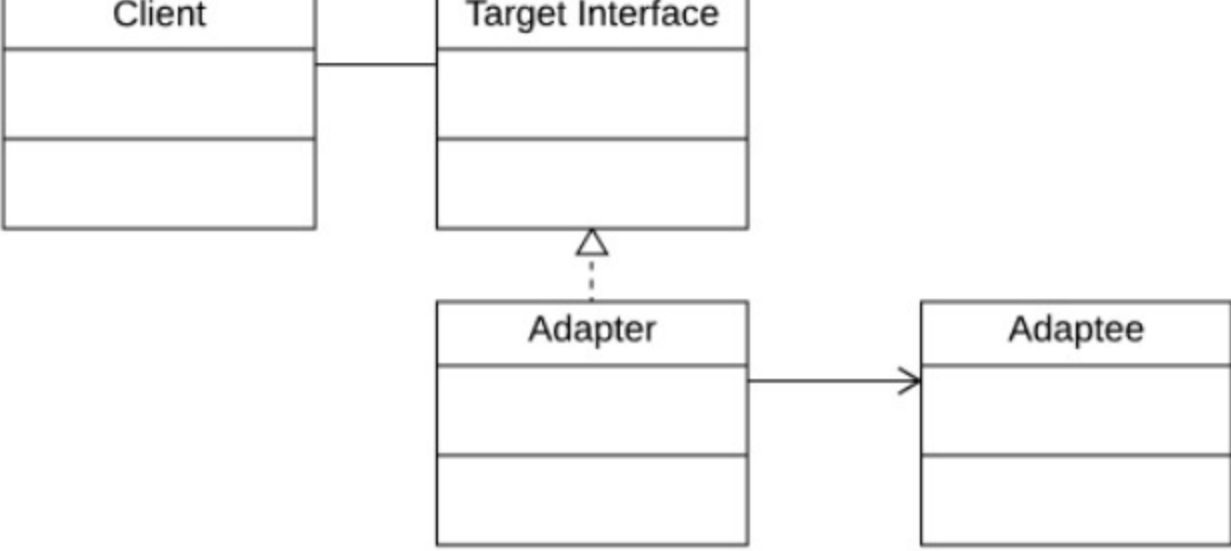
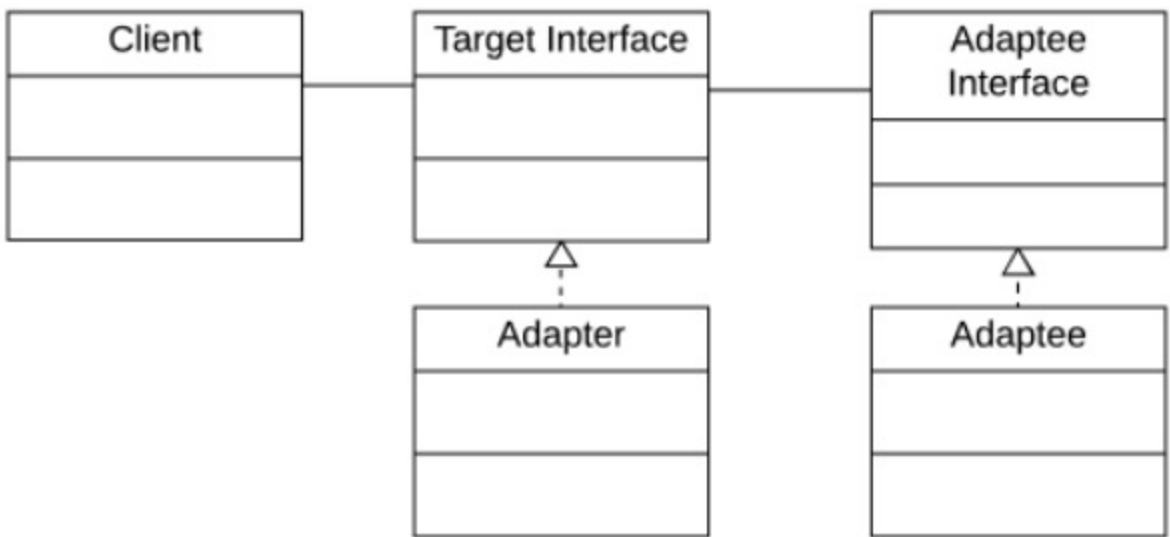
Correct! If the subsystem or client are changed, there are fewer connections to manage.
- ☒ The complexity of the subsystem is hidden
- ✔ Правильно

Correct! The Facade presents a simplified interface to clients.
- ☒ The Facade class redirects requests as needed
- ✔ Правильно

Correct! This is one of the ways that the Facade can simplify for the client.
- ☐ The subsystem can handle more clients

6. Which of the following diagrams shows the **Adapter** pattern?

1 / 1 балл

- a) 
- b) 
- c) 
- d) 
- ☐ d)
- ☐ b)
- ☐ a)
- ☒ c)
- ✔ Правильно

Correct! The adapter wraps the adaptee and provides its functionality as a target interface that the client can connect with.

7. Which of these are the best applications for a **Composite** Pattern? Choose the **three correct** answers.

1 / 1 балл

- ☒ Files and folders
- ✔ Правильно

Correct! Folders (composite class) can contain other folders, or files (leaf class)
- ☒ Elements in a user-interface dialog
- ✔ Правильно

Correct! Elements in a dialog may contain other elements (a composite class) or they may not (leaf class)
- ☐ Students in a class
- ☒ Music in a playlist
- ✔ Правильно

Correct! Each playlist can be composed of songs or other playlists -- or a combination of both.

8. Which of these is **NOT** a common application of the **Proxy** Pattern?

0 / 1 балл

- ☒ virtual proxy
- ☐ remote proxy
- ☐ information proxy
- ☐ protection proxy
- ✘ Неправильно

Incorrect. Virtual proxies are used when the subject is resource intensive. Try again!

9. How does a **Decorator** Pattern work? Choose one.

1 / 1 балл

- ☒ builds a behaviour by stacking objects
- ☐ expands the methods of a class with inheritance
- ☐ adding features to a class with a new class
- ☐ encapsulates a class to give it a different interface
- ✔ Правильно

Correct! This accurately describes a Decorator pattern.

10. What are the object types that are used in the **Composite** Pattern? Select the **two correct** answers.

1 / 1 балл

- ☒ leaf
- ✔ Правильно

Correct! A leaf is the term for a composite subclass that cannot contain another component
- ☐ branch
- ☐ root
- ☒ composite
- ✔ Правильно

Correct. A composite object is a component object that can contain other components, instances of either other composites, or leaf classes.
- ☐ trunk

11. Many different clients need to create a similar object. You would like to outsource this concrete instantiation to a dedicated class. Which technique will you use, in one word?

0 / 1 балл

abstract

✘ Неправильно

12. How do you enforce the creation of only one Singleton object? Select the **two correct** answers.

1 / 1 балл

- ☐ Specify in the comments that only one Singleton object is to be instantiated.
- ☒ Write a method that can create a new Singleton object or return the existing one.
- ✔ Правильно

Correct! If the Singleton class is already instantiated, simply return that object. If it doesn't, make it!
- ☐ Throw an exception if a Singleton object is already instantiated
- ☒ Give the Singleton class a private constructor
- ✔ Правильно

Correct. This essentially only allows the Singleton to construct itself, which it will not do if it is already instantiated once.