Sure, let's break down each method in detail and provide a description for each one:

**TestRequestBuilderClass.java:**

1. **private TestRequestBuilderClass()**
   * Description: This is a private constructor of the **TestRequestBuilderClass**. It is used to initialize the **RequestSpecification** object (**requestSpec**) when an instance of **TestRequestBuilderClass** is created.
   * Purpose: Initializes the **RequestSpecification** object.
2. **public RequestSpecification getRequestSpec()**
   * Description: This method returns the configured **RequestSpecification** object.
   * Purpose: Allows external classes to access the configured **RequestSpecification** object.
3. **public static class Builder**
   * Description: This is a nested static class **Builder** inside **TestRequestBuilderClass**. It is used to facilitate the builder pattern for creating instances of **TestRequestBuilderClass**.
   * Purpose: Provides a fluent interface for configuring and building instances of **TestRequestBuilderClass**.
4. **public Builder()**
   * Description: This is the constructor of the **Builder** class. It initializes an instance of **TestRequestBuilderClass**.
   * Purpose: Initializes an instance of **TestRequestBuilderClass** for configuration.
5. **public Builder setBaseURI(String baseURI)**
   * Description: This method sets the base URI for the request.
   * Parameters:
     + **baseURI**: The base URI to be set for the request.
   * Returns: The same **Builder** object for method chaining.
   * Purpose: Sets the base URI for the request.
6. **public Builder setQueryParam(String paramName, Object value)**
   * Description: This method sets a query parameter for the request.
   * Parameters:
     + **paramName**: The name of the query parameter.
     + **value**: The value of the query parameter.
   * Returns: The same **Builder** object for method chaining.
   * Purpose: Sets a query parameter for the request.
7. **public TestRequestBuilderClass build()**
   * Description: This method builds and returns an instance of **TestRequestBuilderClass** with the configured parameters.
   * Returns: An instance of **TestRequestBuilderClass** configured with the parameters set using the builder methods.
   * Purpose: Constructs and returns an instance of **TestRequestBuilderClass** with the configured parameters.

**Ch12BuilderPatternExample.java:**

1. **@Test(description = "Get user by ID")**
   * Description: This is a TestNG test method annotated with **@Test**. It represents the test scenario for getting a user by ID.
   * Purpose: Executes the test scenario.
2. **public void testGetUserById()**
   * Description: This method defines the test scenario for getting a user by ID.
   * Steps:
     1. Create an instance of **TestRequestBuilderClass** using the builder pattern, setting the base URI and query parameters.
     2. Send a GET request using the configured request specification.
     3. Extract the response.
     4. Validate the response status code.
     5. Print the API response.
   * Purpose: Executes the test scenario for getting a user by ID.

