The code is an example of custom serialization, it's a class called HtmSerializer which contains several methods that handle the serialization and deserialization of different types. The methods include:

1. SerializeBegin(string typeName, StreamWriter sw) - This method is used to write the start marker of the type that is being serialized.
2. ReadBegin(string typeName) - This method reads the start marker of the type that is being serialized.
3. SerializeEnd(string typeName, StreamWriter sw) - This method writes the end marker of the type that is being serialized.
4. ReadEnd(string typeName) - This method reads the end marker of the type that is being serialized.
5. SerializeValue(int val, StreamWriter sw) - This method writes the value of an integer type property.
6. Serialize(object obj, object parent, StreamWriter sw) - This method serializes the object to the stream.
7. Save(string fileName, object obj) - This method saves the object to the file.
8. Load<T>(string fileName) - This method loads the object from the file.
9. TryLoad<T>(string fileName, out T obj) - This method tries to load the object from the file and returns true if the file exists, false otherwise.

**It seems the class is handling serialization for simple types like int and also for complex objects, this is done by calling the Serialize method recursively on the properties of the object, and it's also keeping track of the HashCodes of the objects that are being serialized to handle circular references.**

**From line 219 🡪 444**

1. SerializeKeyValuePair(string name, object obj, StreamWriter sw) - This method is used to serialize KeyValuePair objects. It uses reflection to get the fields of the KeyValuePair object and calls the Serialize method recursively on the key and value properties.
2. Serialize1(object obj, string name, StreamWriter sw, Dictionary<Type, Action<StreamWriter, string, object>> customSerializers = null) - This method is used to serialize different types of objects. It calls different methods based on the type of the object, such as SerializeBegin, SerializeValue, SerializeDictionary, SerializeIEnumerable, SerializeObject, etc.
3. SerializeDictionary(string name, object obj, StreamWriter sw, List<string> ignoreMembers = null) - This method is used to serialize Dictionary objects. It iterates through the items of the dictionary and calls the Serialize method recursively on the key and value properties.
4. SerializeBegin(string propName, StreamWriter sw, Type type) - This method is used to write the start marker of the object or the property being serialized.
5. SerializeEnd(string propName, StreamWriter sw, Type type) - This method is used to write the end marker of the object or the property being serialized.

**It looks like the class is using a StreamWriter to output the serialized data and it's also using reflection to get the properties and fields of the objects that are being serialized.**