Typescript. Task 2

Create one base class and two derived classes. Realize two interfaces. Derived classes have to implement this interfaces (first – one interface, second – two interfaces). Override one method of the base class in one of derived classes. Create an array of objects and demonstrate polymorphism principle on it (call overridden method in for loop).

Option 1. Base class is abstract and has abstract method. Method of the base class that has to be overridden accepts string parameter and returns boolean. One of the interfaces has two fields: number and string, one of which is readonly. An array of objects has base class data type.

Option 2. Base class is abstract but doesn't have abstract methods. Method of the base class that has to be overridden accepts two number parameters and returns nothing. One of the interfaces has tree fields: number, boolead and string, one of which is optional. An array of objects has one of interfaces data type.

Option 3. Base class is not abstract and array of objects has elements with the type of base class. Method of the base class that has to be overridden accepts boolean and string parameters and returns number. One of the interfaces has one field: number, which is readonly. An array of objects has base class data type.

Option 4. Base class is abstract and has abstract method. Method of the base class that has to be overridden accepts nothing and returns enum value. One of the interfaces has two fields: number and boolead, one of which is readonly. An array of objects has one of interfaces data type.

Option 5. Base class is abstract but doesn't have abstract methods. Method of the base class that has to be overridden accepts array of numbers parameter and

returns number. One of the interfaces has four fields: number, number, string and boolean, all of them are readonly. An array of objects has base class data type.

Option 6. Base class is not abstract and array of objects has elements with the type of base class. Method of the base class that has to be overridden accepts two string parameters and returns array of strings. One of the interfaces has two fields: boolean and number, one of which is optional. An array of objects has one of interfaces data type.

Option 7. Base class is abstract and has abstract method. Method of the base class that has to be overridden accepts string and boolean parameters and returns boolean. One of the interfaces has three fields: number, string and string, one of which is optional. An array of objects has base class data type.

Option 8. Base class is abstract but doesn't have abstract methods. Method of the base class that has to be overridden accepts number parameter and returns number. One of the interfaces has two fields: number and number, one of which is readonly and another is optional. An array of objects has one of interfaces data type.

Option 9. Base class is not abstract and array of objects has elements with the type of base class. Method of the base class that has to be overridden accepts string and boolean parameters and returns array of boolean. One of the interfaces has three fields: all numbers, two of which is optional. An array of objects has base class data type.

Option 10. Base class is abstract and has abstract method. Method of the base class that has to be overridden accepts rest number parameter and returns array of numbers. One of the interfaces has four fields: two numbers and two strings, one of which is readonly. An array of objects has one of interfaces data type.