Practice 11

Deadline: 2 weeks from now. Should be checked onsite (during labs).

In application development, developers typically need to performance validation checking, e.g., check the validity of username and password. To avoid writing redundant logics, you, as a framework developer, provide a framework code that takes care of all boilerplate logics, so that client developers, by using your framework, could use annotations to simplify the data validation.

Client Code

```
public class User {

    @MinLength
    @CustomValidation(rule = Rule.ALL_LOWERCASE)
    private String username;


    @MinLength(min=8)
    @CustomValidation(rule = Rule.NO_USERNAME)
    @CustomValidation(rule = Rule.HAS_BOTH_DIGITS_AND_LETTERS)
    private String password;


    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }

}
```

Framework Code

Your task is to finish the framework code, which supports the following annotations to annotate fields:

- `@MinLength`: the annotated field's minimum lenth. Default is 3.
- `@CustomValidation`: which supports 3 rules (see above code) on fields. You may use a `enum` type for `Rule`. In addition, this annotation can be used multiple times on a field. Check this webpage for how to use the `@Repeatable` meta annotation for this purpose.

Your framework code should also use reflection to implement all the validation logics.

If your implementation is correct, then executing the client code together with your framework code should expect the following results:

```
public class DataValidator {
```

```
    public static boolean validate(Object obj) {
        // TODO
    }


    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while(true){
            System.out.print("Username: ");
            String username = sc.next();
            System.out.print("Password: ");
            String pwd = sc.next();

            User user = new User(username,pwd);
            if(validate(user)){
                System.out.println("Success!");
                break;
            }
        }
    }
}
```

```
Username: a1
Password: 123
Validation failed for field *username*: should have a minimum length of 3
Validation failed for field *password*: should have a minimum length of 8
Validation failed for field *password*: should have both letters and digits
Username: ABC
Password: 12345678
Validation failed for field *username*: should be all lowercase
Validation failed for field *password*: should have both letters and digits
Username: abc
Password: 12345abc
Validation failed for field *password*: should not contain username
Username: abc
Password: 12345ABCD
Success!
```

Evaluation

The practice will be checked by teachers or SAs. What will be tested:

1. That you understand every line of your own code, not just copy from somewhere
2. That your program compiles correctly (javac)
3. Correctness of the program logic
4. That the result is obtained in a reasonable time

Late submissions after the deadline will incur a 20% penalty, meaning that you can only get 80% of this practice's score.