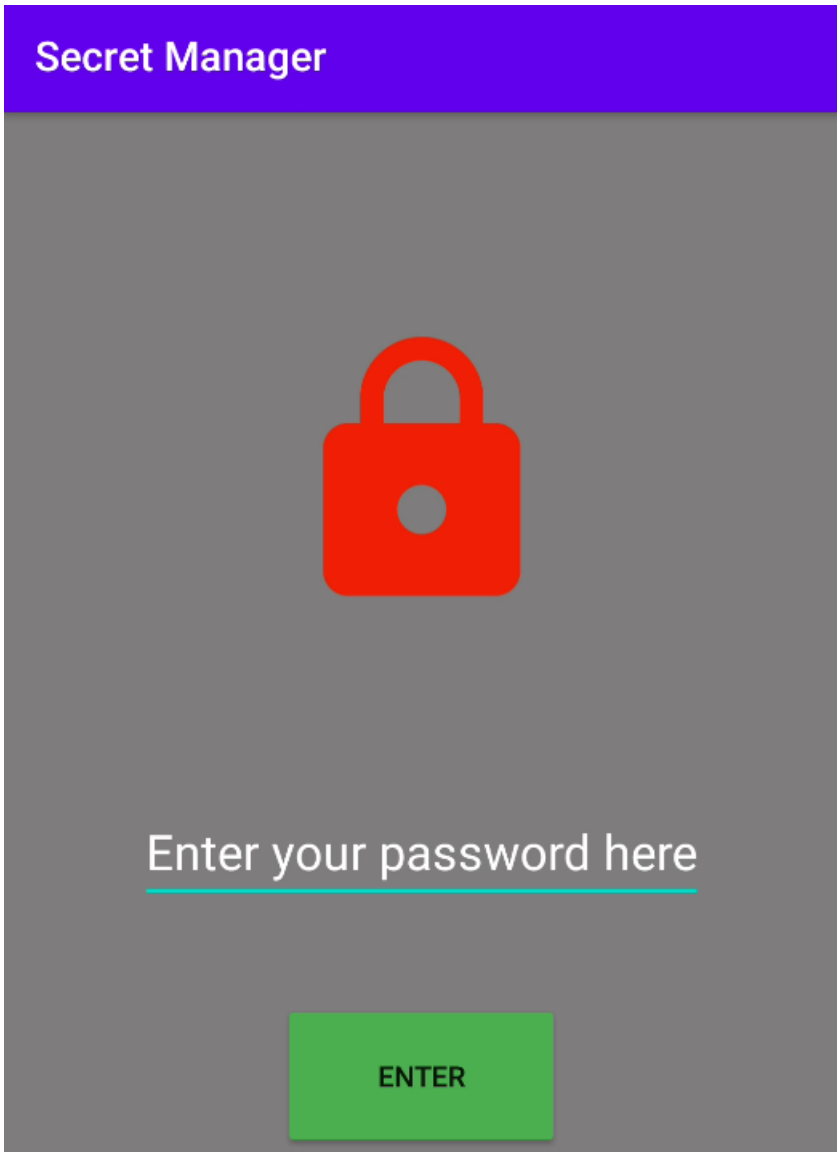
A. Secret Manager

1. Diberikan sebuah file apk, dimana kita diminta memasukkan password



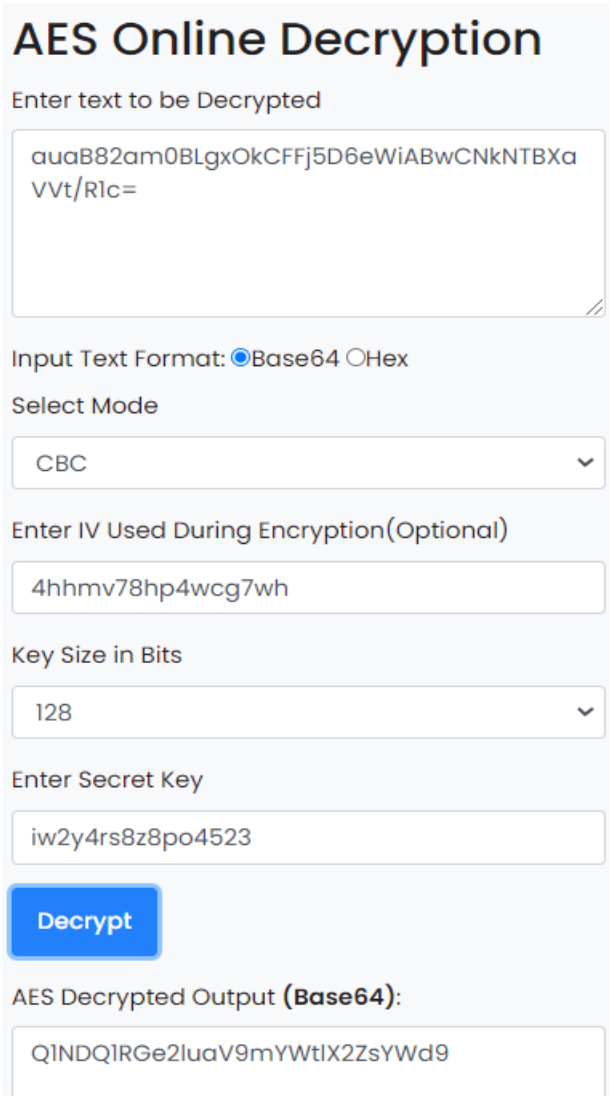
2. Decompile aplikasinya, disitu ada validasi, jika inputan user sama dengan Secret.decrypt("7v+JKXCkLHX46gIpx2EiEg=="), maka flag bisa diakses di logcat

```
31 public class MainActivity extends AppCompatActivity {
       private EditText editTextpassword;
       Secret s = new Secret();
       private Button submit;

       /* access modifiers changed from: protected */
       @Override // androidx.activity.ComponentActivity, androidx.core.app.ComponentActivity, androidx.appcompat.app.AppCompatActivity, androidx.fragment.app.FragmentActivity
32     public void onCreate(Bundle bundle) {
33         super.onCreate(bundle);
34         setContentView(R.layout.activity_main);
35         this.submit = (Button) findViewById(R.id.submit);
36         this.editTextpassword = (EditText) findViewById(R.id.editTextPassword);
37         this.submit.setOnClickListener(new View.OnClickListener() {
               /* class com.chevaliers.secretmanager.MainActivity.AnonymousClass1 */

39             public void onClick(View view) {
41                 if (MainActivity.this.editTextpassword.getText().toString().trim().equals(Secret.decrypt("7v+JKXCkLHX46gIpx2EiEg==", ))) {
42                     Log.d("flag", "Flag : " + Secret.decrypt(MainActivity.this.getResources().getString(R.string.flag)));
                   }
               }
           });
48         registerReceiver(this.s, new IntentFilter("com.chevaliers.secretmanager.secret.GET_SECRET"));
       }
   }
```

3. Jika kita lihat class Secret, disitu method decrypt() menggunakan algoritma AES. Karena ada key dan IV, kita bisa dengan mudah dekrip AESnya

```
24 public class Secret extends BroadcastReceiver {
       private static final String key = "iw2y4rs8z8po4523";
       private static final String myiv = "4hhmv78hp4wcg7wh";

25     public static String encrypt(String str) {
           try {
26             IvParameterSpec ivParameterSpec = new IvParameterSpec(myiv.getBytes("UTF-8"));
27             SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
29             Cipher instance = Cipher.getInstance("AES/CBC/PKCS7PADDING");
30             instance.init(1, secretKeySpec, ivParameterSpec);
33             return Base64.encodeToString(instance.doFinal(str.getBytes()), 0);
           } catch (Exception e) {
35             e.printStackTrace();
35             return null;
           }
       }

41     public static String decrypt(String str) {
           try {
42             IvParameterSpec ivParameterSpec = new IvParameterSpec(myiv.getBytes("UTF-8"));
43             SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
45             Cipher instance = Cipher.getInstance("AES/CBC/PKCS5PADDING");
46             instance.init(2, secretKeySpec, ivParameterSpec);
48             return new String(instance.doFinal(Base64.decode(str, 0)));
           } catch (Exception e) {
51             e.printStackTrace();
51             return null;
           }
       }
```

4. Karena flagnya bisa diakses di strings.xml, maka kita tidak perlu menggunakan logcat. Akses strings.xml, dan disitu ada flag dengan value auaB82am0BLgxOkCFFj5D6eWiABwCNkNTBXaVVt/R1c=. Sepertinya flag dienkrip dengan AES dan key serta IV diatas. Gunakan AES Online Decryption, dan kita berhasil dekrip cipherteksnya dalam format base64.



5. Decode dengan base64, ternyata fake flag :(

```
┌─[chevaliers@parrot]─[~/Desktop/ctf/cscctf/probset/re/SecretManager]
└──╼ $echo 'Q1NDQ1RGe2luaV9mYWtlX2ZsYWd9' | base64 -d
CSCCTF{ini_fake_flag}┌─[chevaliers@parrot]─[~/Desktop/ctf/cscctf/probset/re/SecretManager]
└──╼ $
```

6. Jika kita perhatikan lagi Class Secret, disitu Class Secret inherit Class BroadcastReceiver, sehingga mungkin kita perlu exploit BroadcastReceiver untuk memperoleh flag. Disitu jika ada broadcast intent dengan action com.chevaliers.secretmanager.secret.GET_SECRET, maka broadcast intent akan diterma dan ada logic untuk memanggil sebuah data dari firebase. Data tersebut akan dipassing sebagai argument untuk method f()

```
58      public void onReceive(Context context, Intent intent) {
59          if ("com.chevaliers.secretmanager.secret.GET_SECRET".equals(intent.getAction())) {
69              FirebaseFirestore.getInstance().collection("c").document("d").get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
                    /* class com.chevaliers.secretmanager.secret.Secret.AnonymousClass2 */

64                  public void onSuccess(DocumentSnapshot documentSnapshot) {
65                      if (documentSnapshot.exists()) {
66                          Secret.this.f(documentSnapshot);
                        }
                    }
                }).addOnFailureListener(new OnFailureListener() {
                    /* class com.chevaliers.secretmanager.secret.Secret.AnonymousClass1 */

                    @Override // com.google.android.gms.tasks.OnFailureListener
71                  public void onFailure(Exception exc) {
72                      Log.d("error", "onFailure: " + exc.toString());
                    }
                });
            }
        }
```

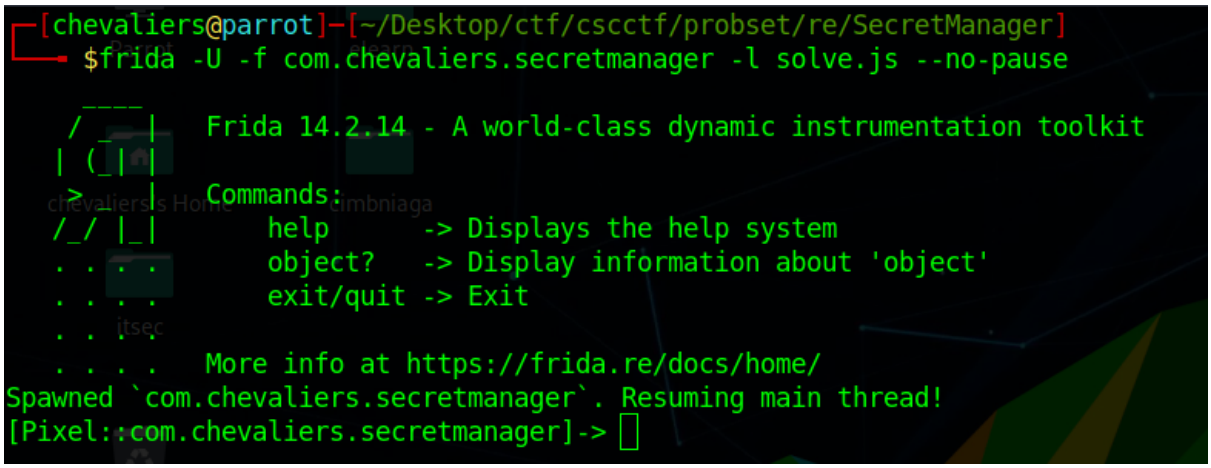
7. Jika kita lihat method f, disitu akan return documentSnapshot.toString()

```
77      public String f(DocumentSnapshot documentSnapshot) {
78          return documentSnapshot.toString();
        }
    }
```

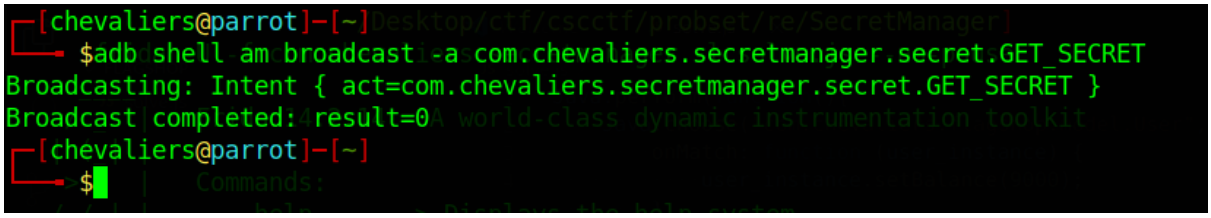
8. Karena method f tidak pernah dipanggil oleh activity manapun, maka kita perlu buat sebuah frida script untuk hook method f, dan ambil parameternya. Codenya adalah seperti dibawah :

```
Java.perform(function(){

    var hook = Java.use("com.chevaliers.secretmanager.secret.Secret");


hook.f.overload("com.google.firebase.firestore.DocumentSnapshot").imple
mentation = function(a){

        console.log(a.toString());

        return a.toString();

    }

});
```

9. Panggil script fridanya

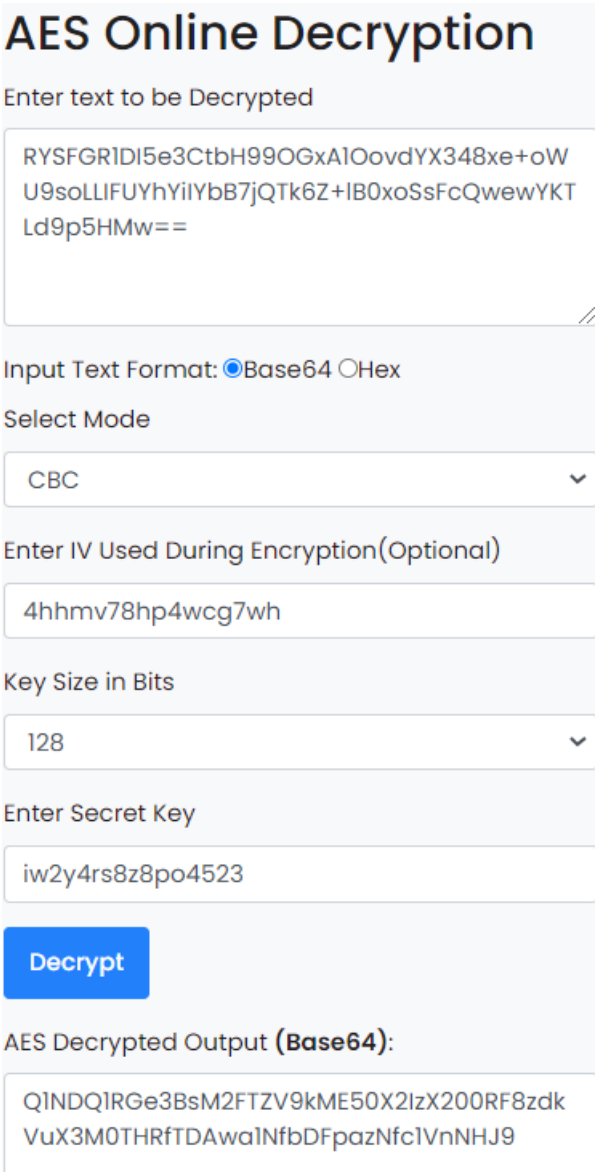


10. Kirim broadcast intent dengan action com.chevaliers.secretmanager.secret.GET_SECRET, commandnya **adb shell am broadcast -a com.chevaliers.secretmanager.secret.GET_SECRET**



11. Setelah itu frida berhasil mengcapture data dari firebase. Datanya sepertinya dienkrip dengan AES. Kita coba gunakan key dan IV yang diatas.

12. Dekrip lagi dengan AES Online Decryption

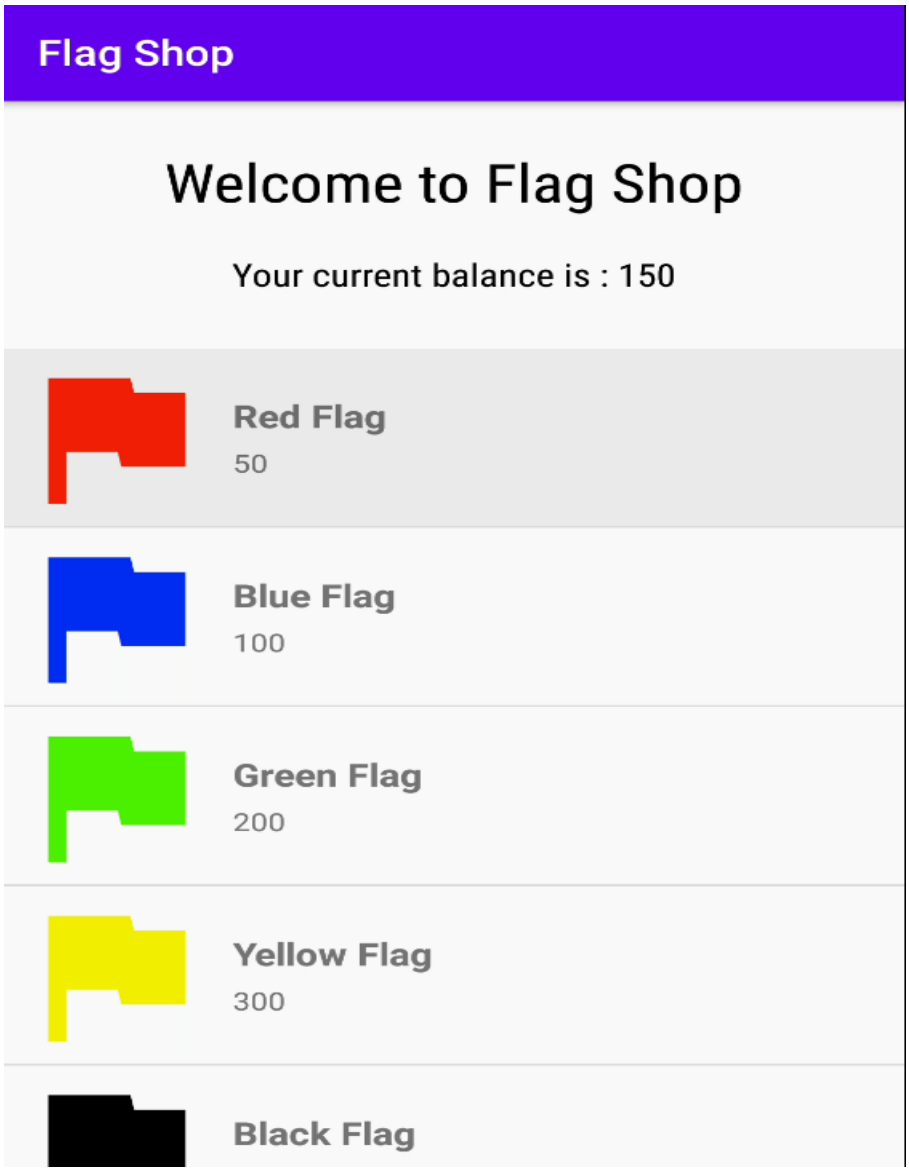


13. Lalu decode base64nya, dan kita berhasil mendapatkan flag

```
┌─[chevaliers@parrot]─[~/Desktop/ctf/cscctf/probset/re/SecretManager]
└──╼ $echo 'Q1NDQ1RGe3BsM2FTZV9kME50X2IzX200RF8zdkVuX3M0THRfTDAwa1NfbDFpazNfc1VnNHJ9' | base64 -d
CSCCTF{pl3aSe_d0Nt_b3_m4D_3vEn_s4Lt_L00kS_l1ik3_sUg4r}┌─[chevaliers@parrot]─[~/Desktop/ctf/cscctf/probset/re/SecretManager]
└──╼ $
```

B. Flag Shop

1. Diberikan sebuah soal android dimana disitu terdapat flag shop. Uang yang pertama kali akan diperoleh user adalah 150, sehingga flag yang bisa dibeli hanyalah Red Flag dan Blue Flag.



2. Decompile aplikasi dengan jadx-gui. Pada Class MainActivity, disitu terdapat instance dari Class User bernama user. Instance user akan memanggil method User.getInstance()

```
31 public class MainActivity extends AppCompatActivity {
       private int[] images = {R.drawable.red_flag, R.drawable.blue_flag, R.drawable.green_flag, R.drawable.yellow_flag, R.drawable.black_flag};
       private String[] item_name = {"Red Flag", "Blue Flag", "Green Flag", "Yellow Flag", "Black Flag"};
       private ListView list;
       private int[] price = {50, 100, LogSeverity.INFO_VALUE, LogSeverity.NOTICE_VALUE, LogSeverity.ERROR_VALUE};
       private User user;
       private TextView wallet;

       /* access modifiers changed from: protected */
       @Override // androidx.activity.ComponentActivity, androidx.core.app.ComponentActivity, androidx.appcompat.app.AppCompatActivity, androidx.fragment.app.FragmentActivity
32     public void onCreate(Bundle bundle) {
33         super.onCreate(bundle);
34         setContentView(R.layout.activity_main);
35         this.user = User.getInstance();
36         this.list = (ListView) findViewById(R.id.list);
37         TextView textView = (TextView) findViewById(R.id.wallet_textView);
37         this.wallet = textView;
38         textView.setText("Your current balance is : " + this.user.getBalance());
40         this.list.setAdapter((ListAdapter) new MyAdapter(this, this.item_name, this.price, this.images));
41         this.list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
               /* class com.chevaliers.flagshop.MainActivity.AnonymousClass1 */

               @Override // android.widget.AdapterView.OnItemClickListener
43             public void onItemClick(AdapterView<?> adapterView, View view, int i, long j) {
44                 Intent intent = new Intent(MainActivity.this.getApplicationContext(), ItemDetail.class);
45                 intent.putExtra(FirebaseAnalytics.Param.ITEM_NAME, MainActivity.this.item_name[i]);
46                 intent.putExtra("item_price", MainActivity.this.price[i]);
47                 intent.putExtra("item_image", MainActivity.this.images[i]);
49                 MainActivity.this.startActivity(intent);
               }
           });
       }
   }
```

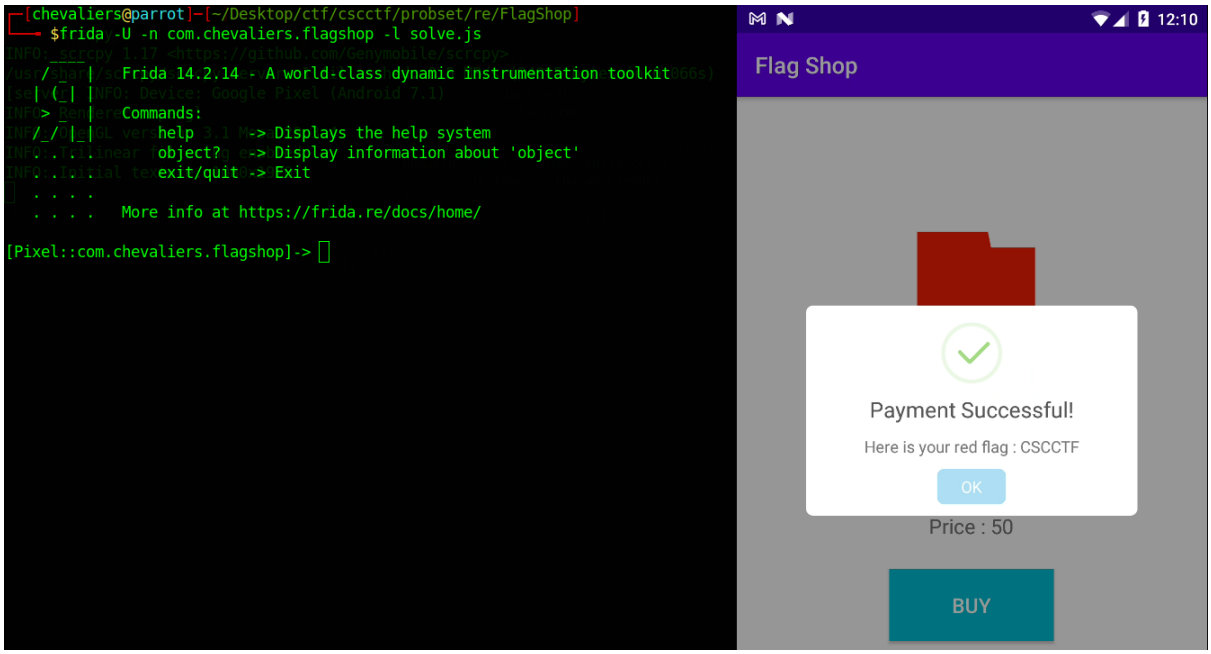
3. Jika kita lihat method getInstance() pada Class User, disitu ada logic singleton. Jika instance dari Class User bernilai null, maka getInstance() akan buat object dengan name "User" dan balancenya 150.

```
27     public static User getInstance() {
28         if (instance == null) {
29             instance = new User("User", 150);
           }
31         return instance;
       }
   }
```

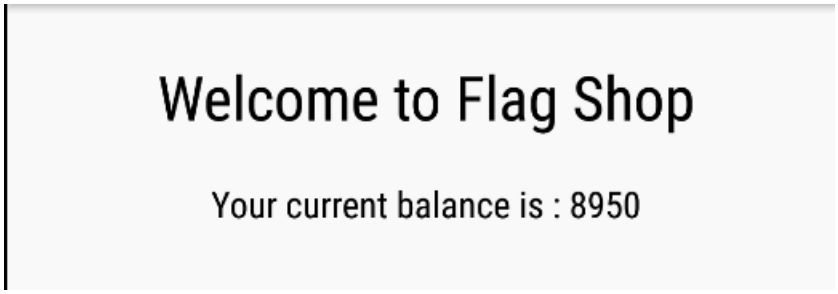
4. Untuk overwrite balance, kita dapat gunakan frida dengan mencari instance dari Class User. Jika ada, maka kita akan panggil setBalance(9000) pada instance tersebut, sehingga uang kita akan menjadi 9000.

```
Java.choose("com.chevaliers.flagshop.model.User", {
        onMatch: function (user_instance) {
            user_instance.setBalance(9000);
        },
        onComplete: function () { }

    });
```

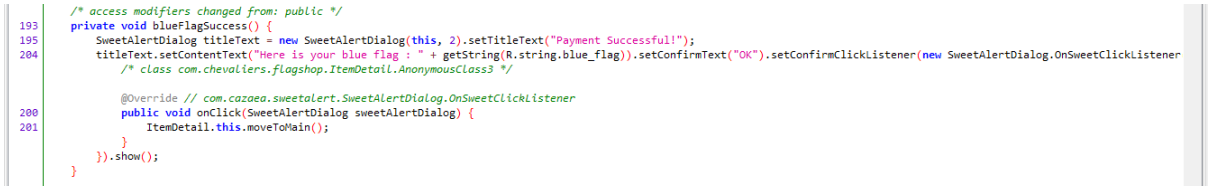
5. Jalankan script fridanya dan beli red flag. Setelah itu, kita dapat bagian pertama dari flag, yaitu CSCCTF



6. Saat klik OK, kita akan kembali ke MainActivity. Disitu balance kita akan menjadi 8950, karena kita sudah setBalance menjadi 9000, dan membeli flag seharga 50, sehingga balance kita menjadi 8950.

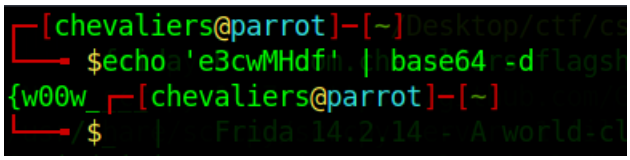


7. Selanjutnya, untuk mempermudah, kita lihat method dari blueFlagSuccess(). Disitu jika berhasil beli flag, maka akan ditampilkan string yang dapat diperoleh di strings.xml, dengan name blue_flag.



```
    /* access modifiers changed from: public */
193 private void blueFlagSuccess() {
195     SweetAlertDialog titleText = new SweetAlertDialog(this, 2).setTitleText("Payment Successful!");
204     titleText.setContentText("Here is your blue flag : " + getString(R.string.blue_flag)).setConfirmText("OK").setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener
            /* class com.chevaliers.flagshop.ItemDetail.AnonymousClass3 */

            @Override // com.cazaea.sweetalert.SweetAlertDialog.OnSweetClickListener
200         public void onClick(SweetAlertDialog sweetAlertDialog) {
201             ItemDetail.this.moveToMain();
            }
        }).show();
    }
```

8. Jika kita ke strings.xml, disitu value dari blue_flag adalah **e3cwMHdf**. Kita coba untuk decode string tersebut dengan base64, dan disitu kita dapat bagian kedua dari flag, yaitu {w00w_, sehingga jika disatukan akan menjadi : CSCCTF{w00w_



9. Lihat kembali source code. Disitu nilai dari green flag akan diambil dari firebase. Syarat mendapatkan green flag adalah minimal balance adalah 200, dan syarat tersebut sudah terpenuhi. Syarat kedua, Flag.getInstance() tidak boleh null.

```
} else if (balance < 0 || Flag.getInstance() == null) {
    ItemDetail.this.failed();
} else {
    FirebaseFirestore.getInstance().collection("c").document("d").get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
        /* class com.chevaliers.flagshop.ItemDetail.AnonymousClass1.AnonymousClass2 */

        public void onSuccess(DocumentSnapshot documentSnapshot) {
            if (documentSnapshot.exists()) {
                SweetAlertDialog titleText = new SweetAlertDialog(ItemDetail.this, 2).setTitleText("Payment Successful!");
                titleText.setContentText("Here is your green flag : " + documentSnapshot.getString("g")).setConfirmText("OK").setConfirmClickListener(new Swee
                    /* class com.chevaliers.flagshop.ItemDetail.AnonymousClass1.AnonymousClass2.AnonymousClass1 */

                    @Override // com.cazaea.sweetalert.SweetAlertDialog.OnSweetClickListener
                    public void onClick(SweetAlertDialog sweetAlertDialog) {
                        ItemDetail.this.moveToMain();
                    }
                }).show();
            }
        }
```

10. Jika kita ke class Flag, disitu ada constructor dengan parameter String, boolean, dan int. Lalu ada singleton pada method getInstance(), dimana akan ada validasi, jika instance dari Class Flag null, maka akan return null.

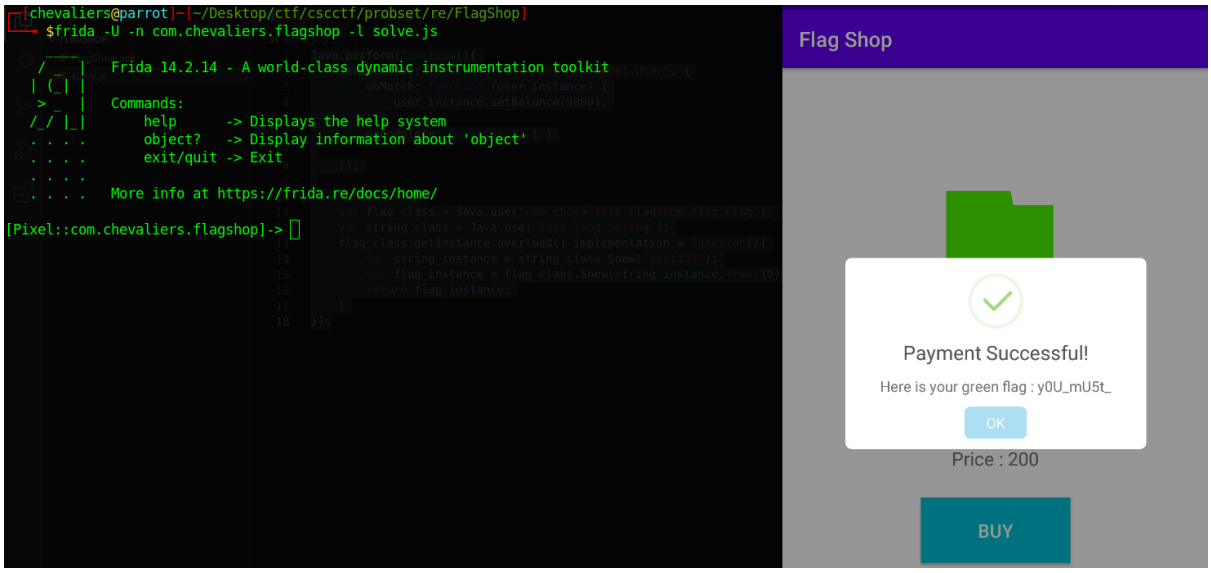
```
 9  public final class Flag {
        private static Flag instance;
        private String property1;
        private boolean property2;
        private int property3;

10      public Flag(String str, boolean z, int i) {
12          this.property1 = str;
13          this.property2 = z;
14          this.property3 = i;
        }

16      public static Flag getInstance() {
17          Flag flag = instance;
17          if (flag == null) {
17              return null;
            }
17          return flag;
        }
```

11. Buat code untuk handle pembuatan instance dari Flag. Karena disitu tidak ada syarat-syarat dari parameter constructor yang harus dipenuhi, maka saya buat stringnya "test123", booleannya bernilai true, dan intnya bernilai 10. Overload method dari getInstance dengan frida, setelah itu buat instance dari Flag, dan return instancenya. Codenya akan menjadi seperti dibawah :

```
Java.perform(function(){
    Java.choose("com.chevaliers.flagshop.model.User", {
        onMatch: function (user_instance) {
            user_instance.setBalance(9000);
        },
        onComplete: function () { }

    });

    //green flag
    var flag_class = Java.use("com.chevaliers.flagshop.flag.Flag");
    var string_class = Java.use("java.lang.String");
    flag_class.getInstance.overload().implementation = function(){
        var string_instance = string_class.$new("test123");
        var flag_instance = flag_class.$new(string_instance,true,10);
        return flag_instance;
```

```
        }
});
```

12. Jalankan kembali frida scriptnya, dan beli green flag. Disitu kita dapat bagian ketiga dari flag, yaitu y0U_mU5t_ sehingga jika kembali disatukan akan menjadi CSCCTF{w00w_ y0U_mU5t_



13. Selanjutnya, untuk membeli yellow flag, syaratnya balance harus cukup, Flag.getInstance() != null, dan Flag.AnotherFlag.getInstance() !=null. 2 syarat sebelumnya sudah terpenuhi, tinggal kita harus membuat Flag.AnotherFlag.getInstance() != null.

```
} else if (balance < 0 || Flag.getInstance() == null || Flag.AnotherFlag.getInstance() == null) {
    ItemDetail.this.failed();
} else {
    FirebaseFirestore.getInstance().collection("c").document("d").get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
        /* class com.chevaliers.flagshop.ItemDetail.AnonymousClass1.AnonymousClass4 */

        public void onSuccess(DocumentSnapshot documentSnapshot) {
            if (documentSnapshot.exists()) {
                SweetAlertDialog titleText = new SweetAlertDialog(ItemDetail.this, 2).setTitleText("Payment Successful!");
                titleText.setContentText("Here is your yellow flag : " + documentSnapshot.getString("y")).setConfirmText("OK").setConfirmClickListener(new
                    /* class com.chevaliers.flagshop.ItemDetail.AnonymousClass1.AnonymousClass4.AnonymousClass1 */

                    @Override // com.cazaea.sweetalert.SweetAlertDialog.OnSweetClickListener
                    public void onClick(SweetAlertDialog sweetAlertDialog) {
                        ItemDetail.this.moveToMain();
                    }
                }).show();
            }
        }
    }).addOnFailureListener(new OnFailureListener() {
        /* class com.chevaliers.flagshop.ItemDetail.AnonymousClass1.AnonymousClass3 */
```

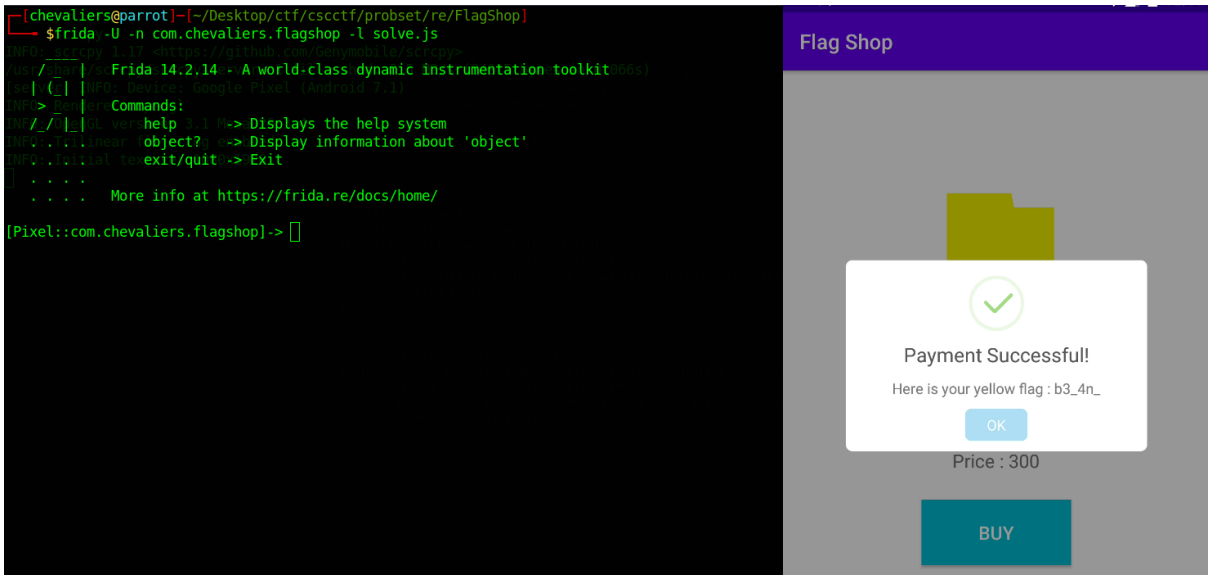
14. AnotherFlag adalah inner class dari Class Flag. Disitu ada constructor dengan parameter String. Lalu ada singleton pada method getInstance(), dimana akan ada validasi, jika instance dari Class AnotherFlag null, maka akan return null.

```
24      public static final class AnotherFlag {
            private static AnotherFlag instance;
            private String property1;

25          public AnotherFlag(String str) {
27              this.property1 = str;
            }

29          public static AnotherFlag getInstance() {
30              AnotherFlag anotherFlag = instance;
30              if (anotherFlag == null) {
30                  return null;
                }
30              return anotherFlag;
            }
```

15. Buat code untuk handle pembuatan instance dari AnotherFlag. Untuk memanggil inner class, kita dapat gunakan <outter class>$<inner class>. Karena disitu tidak ada syarat-syarat dari parameter constructor yang harus dipenuhi, maka saya buat stringnya "test123". Overload method dari getInstance dengan frida, setelah itu buat instance dari AnotherFlag, dan return instancenya. Codenya akan menjadi seperti dibawah :

```
Java.perform(function(){
    Java.choose("com.chevaliers.flagshop.model.User", {
        onMatch: function (user_instance) {
            user_instance.setBalance(9000);
        },
        onComplete: function () { }

    });

    //green flag
    var flag_class = Java.use("com.chevaliers.flagshop.flag.Flag");
    var string_class = Java.use("java.lang.String");
    flag_class.getInstance.overload().implementation = function(){
        var string_instance = string_class.$new("test123");
        var flag_instance = flag_class.$new(string_instance,true,10);
        return flag_instance;
    }

    //yellow flag
    var                    another_flag_class                    =
Java.use("com.chevaliers.flagshop.flag.Flag$AnotherFlag");
    another_flag_class.getInstance.overload().implementation      =
function(){
        var string_instance = string_class.$new("test123");
        var                    another_flag_instance                 =
another_flag_class.$new(string_instance);
        return another_flag_instance;
    }
});
```

16. Jalankan kembali frida scriptnya, dan beli yellow flag. Disitu kita dapat bagian keempat dari flag, yaitu b3_4n_ sehingga jika kembali disatukan akan menjadi CSCCTF{w00w_ y0U_mU5t_b3_4n_



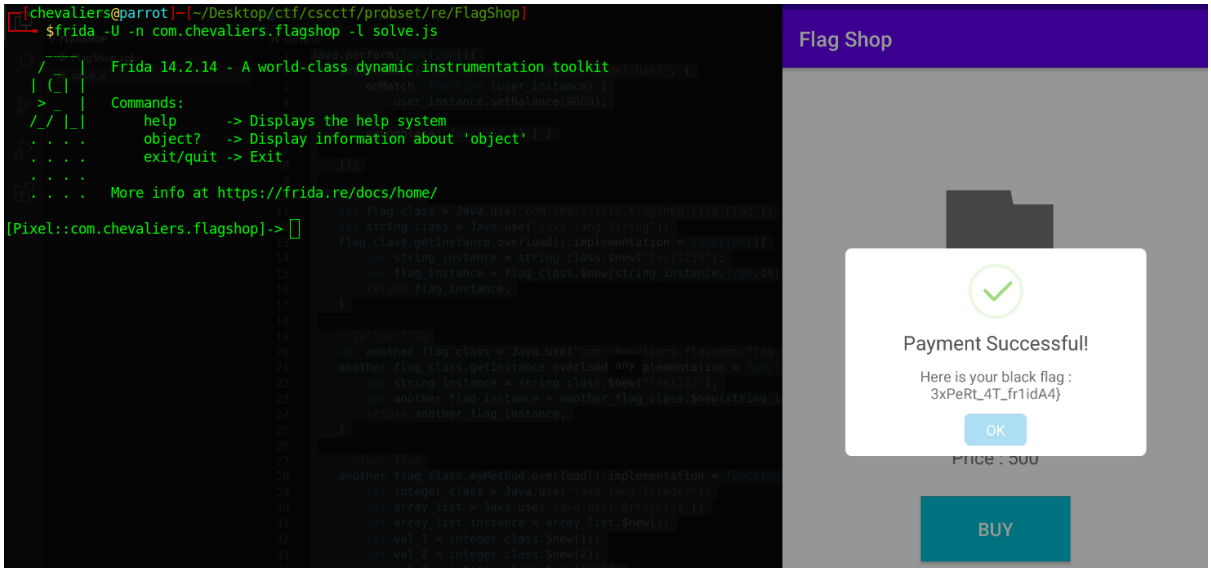
17. Terakhir, untuk beli black flag, ada empat syarat, dimana tiga syarat pertama sama dengan syarat untuk membeli yellow flag. Syarat keempat, method dengan return value ArrayList, dengan nama myMethod(), harus return sebuah ArrayList, dimana index ke-2 nya harus bernilai 1337.

```
if (balance < 0 || Flag.getInstance() == null || Flag.AnotherFlag.getInstance() == null) {
    ItemDetail.this.failed();
} else if (Flag.AnotherFlag.myMethod().get(2).intValue() == 1337) {
    FirebaseFirestore.getInstance().collection("c").document("d").get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
        /* class com.chevaliers.flagshop.ItemDetail.AnonymousClass1.AnonymousClass6 */

        public void onSuccess(DocumentSnapshot documentSnapshot) {
            if (documentSnapshot.exists()) {
                SweetAlertDialog titleText = new SweetAlertDialog(ItemDetail.this, 2).setTitleText("Payment Successful!");
                titleText.setContentText("Here is your black flag : " + documentSnapshot.getString("b")).setConfirmText("OK").setConfirmClickListe
                    /* class com.chevaliers.flagshop.ItemDetail.AnonymousClass1.AnonymousClass6.AnonymousClass1 */

                    @Override // com.cazaea.sweetalert.SweetAlertDialog.OnSweetClickListener
                    public void onClick(SweetAlertDialog sweetAlertDialog) {
                        ItemDetail.this.moveToMain();
                    }
                }).show();
            }
        }
    }).addOnFailureListener(new OnFailureListener() {
        /* class com.chevaliers.flagshop.ItemDetail.AnonymousClass1.AnonymousClass5 */

        @Override // com.google.android.gms.tasks.OnFailureListener
        public void onFailure(Exception exc) {
            ItemDetail.this.failed();
        }
    });
} else {
    ItemDetail.this.failed();
}
```

18. Selanjutnya buat frida script yang akan overload method myMethod. Lalu, buat instance dari Class ArrayList, dan instance dari Class Integer, karena return value dari myMethod adalah ArrayList yang berisi object Integer. Lalu isi ArrayList dengan index 0 => 1, index 1 => 2, index 2 => 1337, karena syarat beli black flag index ke-2 harus 1337. Lalu return ArrayListnya.

```
Java.perform(function(){
    Java.choose("com.chevaliers.flagshop.model.User", {
        onMatch: function (user_instance) {
            user_instance.setBalance(9000);
        },
        onComplete: function () { }

    });

    //green flag
    var flag_class = Java.use("com.chevaliers.flagshop.flag.Flag");
    var string_class = Java.use("java.lang.String");
    flag_class.getInstance.overload().implementation = function(){
        var string_instance = string_class.$new("test1234");
        var flag_instance = flag_class.$new(string_instance,true,10);
        return flag_instance;
    }

    //yellow flag
    var                    another_flag_class                    =
Java.use("com.chevaliers.flagshop.flag.Flag$AnotherFlag");
    another_flag_class.getInstance.overload().implementation      =
function(){
        var string_instance = string_class.$new("test123");
        var                  another_flag_instance                =
another_flag_class.$new(string_instance);
        return another_flag_instance;
    }

    //black flag
    another_flag_class.myMethod.overload().implementation         =
function(){
        var integer_class = Java.use("java.lang.Integer");
        var array_list = Java.use('java.util.ArrayList');
        var array_list_instance = array_list.$new();
        var val_1 = integer_class.$new(1);
        var val_2 = integer_class.$new(2);
        var val_3 = integer_class.$new(1337);
        array_list_instance.add(val_1);
        array_list_instance.add(val_2);
        array_list_instance.add(val_3);
        return array_list_instance;
    }
});
```

20. Jalankan frida script dan beli black flag, dan kita berhasil mendapatkan bagian akhir dari flag. Hasil dari black flag adalah 3xPeRt_4T_fr1idA4}, dan jika digabungkan akan menjadi : CSCCTF{w00w_y0U_mU5t_b3_4n_3xPeRt_4T_fr1idA4}.



21. Final script :

```
Java.perform(function(){
    Java.choose("com.chevaliers.flagshop.model.User", {
        onMatch: function (user_instance) {
            user_instance.setBalance(9000);
        },
        onComplete: function () { }

    });

    //green flag
    var flag_class = Java.use("com.chevaliers.flagshop.flag.Flag");
    var string_class = Java.use("java.lang.String");
    flag_class.getInstance.overload().implementation = function(){
        var string_instance = string_class.$new("test1234");
        var flag_instance = flag_class.$new(string_instance,true,10);
        return flag_instance;
    }

    //yellow flag
    var                    another_flag_class                    =
Java.use("com.chevaliers.flagshop.flag.Flag$AnotherFlag");
    another_flag_class.getInstance.overload().implementation      =
function(){
        var string_instance = string_class.$new("test123");
        var                    another_flag_instance                 =
another_flag_class.$new(string_instance);
        return another_flag_instance;
    }

    //black flag
    another_flag_class.myMethod.overload().implementation = function(){
        var integer_class = Java.use("java.lang.Integer");
        var array_list = Java.use('java.util.ArrayList');
```

```
        var array_list_instance = array_list.$new();
        var val_1 = integer_class.$new(1);
        var val_2 = integer_class.$new(2);
        var val_3 = integer_class.$new(1337);
        array_list_instance.add(val_1);
        array_list_instance.add(val_2);
        array_list_instance.add(val_3);
        return array_list_instance;
    }

});
```