

A step-by-step discussion of Shadow Example

- On the following slides we show in an example how the shadow concept works using a special, modified version of the jikes compiler.
- For theoretical discussion of shadows please see other material that can be found on <http://perisic.com/shadow>.
- So, let's start with a simple class...

```
package com.perisic.jshadow.examples;
public class NiceFellow extends com.perisic.jshadow.Shadowable {
    private String name;
    private boolean isHappy;
    public boolean isHappy() { return isHappy; }
    public void setHappy(boolean what) { isHappy = what; }
    public String getName() { return name; }
    public NiceFellow(String theName) {
        name = theName;
        isHappy = false;
    }
    public void smile(String how) {
        isHappy = true;
        System.out.println(name + " smiles "+how+".");
    }
    public void printMood() {
        if( isHappy() ) { System.out.println(name + " feels happy."); }
        else { System.out.println(name + " feels sad."); }
    }
    public void tellJokeTo(NiceFellow other) {
        System.out.println(name+" says: \"Knock, knock ... \");
        other.smile("happily");
    }
}
```

Example Class: NiceFellow.

The usual suspects:
Attributes, Accessor,
Modifier, Constructor

Smiling

Output depending
on happiness

Interacting with
other objects.

Example of a Shadow (1). A shadow in code is a very normal Java class that extends `com.perisic.jshadow.Shadow`. This class defines the `shadowOwner()` method that returns the shadowed object.

```
package com.perisic.jshadow.examples;
public class SmileShadow extends com.perisic.jshadow.Shadow
{
    public void smile(String how ) {
        shadowOwner().smile("first");
        System.out.println(shadowOwner().getName()+
            " is laughing out loud "+how+"!");
    }
}
```

Example of a Shadow (2). This shadow defines two methods, `tellCalamityTo()` and `cry()`: When an object tells a calamity to another object then the second object cries, and it will not be happy.

```
package com.perisic.jshadow.examples;

public class CalamityShadow extends com.perisic.jshadow.Shadow {

    public void tellCalamityTo(NiceFellow other) {

        System.out.println(((String) shadowOwner().getName()+" says: Oj, oj!");

        other.cry();

    }

    public void cry() {

        System.out.println(shadowOwner().getName()+" cries.");

        ((NiceFellow) shadowOwner()).setHappy(false);

    }

}
```

```
package com.perisic.jshadow.examples;
public class ShadowTester {

    public static void main(String [] args) {
        NiceFellow fabian = new NiceFellow("Fabian");

        fabian.smile("nicely");
        fabian.addShadow(new SmileShadow());
        fabian.smile("nicely");

        com.perisic.jshadow.Util.addShadow(NiceFellow.class, CalamityShadow.class);
        NiceFellow tomislav = new NiceFellow("Tomislav");
        NiceFellow andreas = new NiceFellow("Andreas");

        andreas.tellJokeTo( tomislav );
        tomislav.printMood();

        andreas.tellCalamityTo( tomislav );
        tomislav.printMood();
    }
}
```

And this is the tester to demonstrate the functionality of the shadow concept. (More on the following slides.)

We compile the code with the command:

```
jikes -dy com/perisic/jshadow/examples/*.java
```

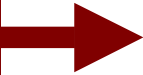
using the version of jikes that is provided on <http://perisic.com/shadow>. That version supports the -dy flag.

Run the code with the command:

```
java com.perisic.jshadow.examples.ShadowTester
```

where java is the “normal” virtual java provided by SUN

This is the output (just in case, you want to know)



Fabian smiles nicely.

Fabian smiles first.

Fabian is laughing out loud nicely!

Andreas says: "Knock, knock ... "

Tomislav smiles happily.

Tomislav feels happy.

Andreas says: Oj, oj!

Tomislav cries.

Tomislav feels sad.

Example 1

```
package com.perisic.jshadow.examples;  
public class ShadowTester {
```

```
    public static void main(String [] args) {  
        NiceFellow fabian = new NiceFellow("Fabian");
```

```
        fabian.smile("nicely");  
        fabian.addShadow(new SmileShadow());  
        fabian.smile("nicely");
```

```
        com.perisic.jshadow.Util.addShadow(NiceFellow.class, CalamityShadow.class);  
        NiceFellow tomlav = new NiceFellow("Tomislav");  
        NiceFellow andreas = new NiceFellow("Andreas");
```

```
        andreas.tellJokeTo( tomlav );  
        tomlav.printMood();
```

```
        andreas.tellCalamityTo( tomlav );  
        tomlav.printMood();  
    }
```

```
}
```

Fabian smiles, no shadow involved so far.

Fabian smiles nicely.

Fabian smiles first.

Fabian is laughing out loud nicely!

Andreas says: "Knock, knock ... "

Tomislav smiles happily.

Tomislav feels happy.

Andreas says: Oj, oj!

Tomislav cries.

Tomislav feels sad.

Example 1

```
package com.perisic.jshadow.examples;
public class ShadowTester {

    public static void main(String [] args) {
        NiceFellow fabian = new NiceFellow("Fabian");

        fabian.smile("nicely");
        fabian.addShadow(new SmileShadow());
        fabian.smile("nicely");

        com.perisic.jshadow.Util.addShadow(NiceFellow.class, CalamityShadow.class);
        NiceFellow tomlav = new NiceFellow("Tomislav");
        NiceFellow andreas = new NiceFellow("Andreas");

        andreas.tellJokeTo( tomlav );
        tomlav.printMood();

        andreas.tellCalamityTo( tomlav );
        tomlav.printMood();
    }
}
```

We add the shadow to the fabian object, and – voila! - the smile method has a different behaviour: The behaviour as it was defined in the SmileShadow.

```
Fabian smiles nicely.
Fabian smiles first.
Fabian is laughing out loud nicely!
Andreas says: "Knock, knock ... "
Tomislav smiles happily.
Tomislav feels happy.
Andreas says: Oj, oj!
Tomislav cries.
Tomislav feels sad.
```

Example 2

```
package com.perisic.jshadow.examples;
public class ShadowTester {

    public static void main(String [] args) {
        NiceFellow fabian = new NiceFellow("Fabian");

        fabian.smile("nicely");
        fabian.addShadow(new SmileShadow());
        fabian.smile("nicely");

        com.perisic.jshadow.Util.addShadow(NiceFellow.class, CalamityShadow.class);
        NiceFellow tomislav = new NiceFellow("Tomislav");
        NiceFellow andreas = new NiceFellow("Andreas");

        andreas.tellJokeTo( tomislav );
        tomislav.printMood();

        andreas.tellCalamityTo( tomislav );
        tomislav.printMood();
    }
}
```

We now apply a shadow to the NiceFellow class: When tomislav and andreas are created, they are shadowed by the CalamityShadow. The fabian object remains unshadowed by the CalamityShadow.

```
Fabian smiles nicely.
Fabian smiles first.
Fabian is laughing out loud nicely!
Andreas says: "Knock, knock ... "
Tomislav smiles happily.
Tomislav feels happy.
Andreas says: Oj, oj!
Tomislav cries.
Tomislav feels sad.
```

Example 2

```
package com.perisic.jshadow.examples;  
public class ShadowTester {
```

```
    public static void main(String [] args) {  
        NiceFellow fabian = new NiceFellow("Fabian");  
  
        fabian.smile("nicely");  
        fabian.addShadow(new SmileShadow());  
        fabian.smile("nicely");
```

```
        com.perisic.jshadow.Util.addShadow(NiceFellow.class, CalamityShadow.class);  
        NiceFellow tomislav = new NiceFellow("Tomislav");  
        NiceFellow andreas = new NiceFellow("Andreas");
```

```
        andreas.tellJokeTo( tomislav );  
        tomislav.printMood();
```

```
        andreas.tellCalamityTo( tomislav );  
        tomislav.printMood();  
    }
```

```
}
```

Nothing exciting happens here.
Andreas tells a joke to Tomislav
and therefore Tomislav is
happy.

```
Fabian smiles nicely.  
Fabian smiles first.  
Fabian is laughing out loud nicely!  
Andreas says: "Knock, knock ... "  
Tomislav smiles happily.  
Tomislav feels happy.  
Andreas says: Oj, oj!  
Tomislav cries.  
Tomislav feels sad.
```

Example 2

```
package com.perisic.jshadow.examples;
public class ShadowTester {

    public static void main(String [] args) {
        NiceFellow fabian = new NiceFellow("Fabian");

        fabian.smile("nicely");
        fabian.addShadow(new SmileShadow());
        fabian.smile("nicely");

        com.perisic.jshadow.Util.addShadow(NiceFellow.class, CalamityShadow.class);
        NiceFellow tomlav = new NiceFellow("Tomislav");
        NiceFellow andreas = new NiceFellow("Andreas");

        andreas.tellJokeTo( tomlav );
        tomlav.printMood();

        andreas.tellCalamityTo( tomlav );
        tomlav.printMood();
    }
}
```

Doesn't look exciting as well – but note that `tellCalamityTo()` and the method `cry()` that is called from within `tellCalamityTo()` were not defined in the original version of `NiceFellow`!

```
Fabian smiles nicely.
Fabian smiles first.
Fabian is laughing out loud nicely!
Andreas says: "Knock, knock ... "
Tomislav smiles happily.
Tomislav feels happy.
Andreas says: Oj, oj!
Tomislav cries.
Tomislav feels sad.
```

Conclusion

- We see here two applications of the shadow concept:
 - Redefining an existing method of a class or object.
 - Extending the functionality of a class or object.
- And the nice thing is:
 - All that can be done at “run-time”.
- For more information see: <http://perisic.com/shadow>