ISEL

Languages and Managed Runtimes

2023

Week 6 – meta-programming

Outline

- Remember – Reflection
- Meta-programming
- Remember – Logger
- LoggerDynamic

Remember - Reflection

Ability to **introspect**, the structure and behavior of a program at **runtime**.

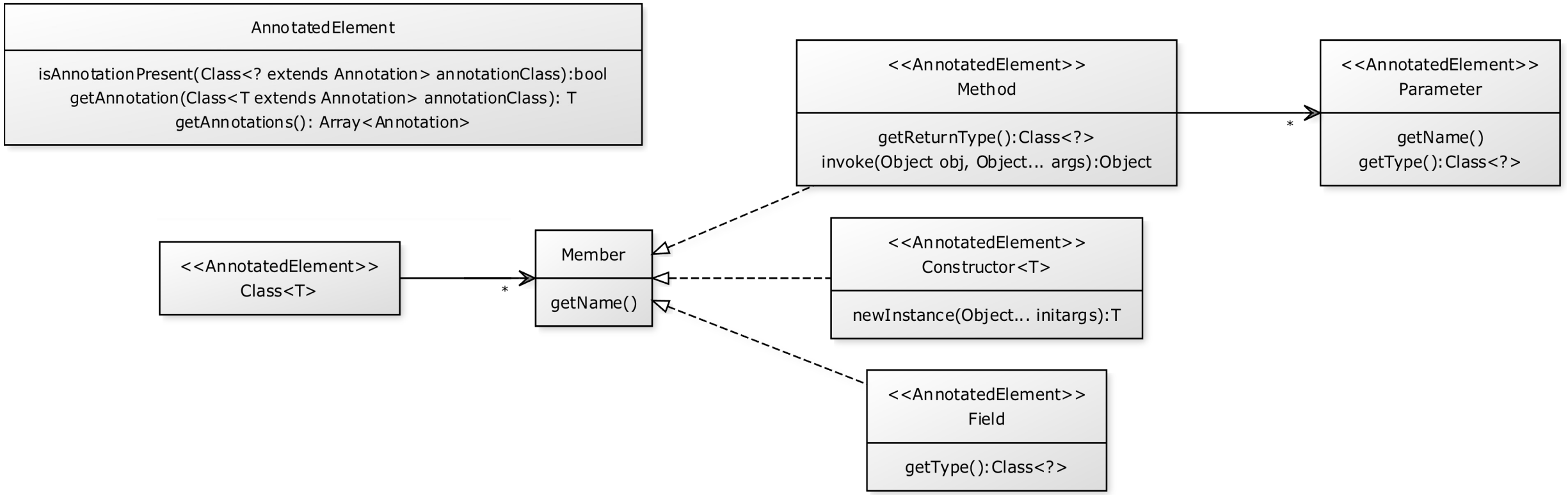
Examples:

- Kotlin - kotlin.reflect
  e.g. `foo::class.declaredFunctions.find{ it.name == "hello"}?.call(foo)`

- Java - java.lang.reflect
  e.g. `foo.getClass().getDeclaredMethod("hello").invoke(foo)`

- JavaScript
  e.g. `foo['hello']()`
- `...`

Java - java.lang.reflect



What can you do?

> E.g. Logger, log4j, Javap, intellisense, Unit tests library (Junit), ORM, dependency injection (Spring), autorouter, etc

What can you **not** do ?

> Modify or **create new Types dynamically** (at runtime).

Outline

- Remember – Reflection
- Meta-programming
- Remember – Logger
- LoggerDynamic

Meta-programming

Ability to read, generate, transform or modify programs **dynamically** (at runtime).

Examples:

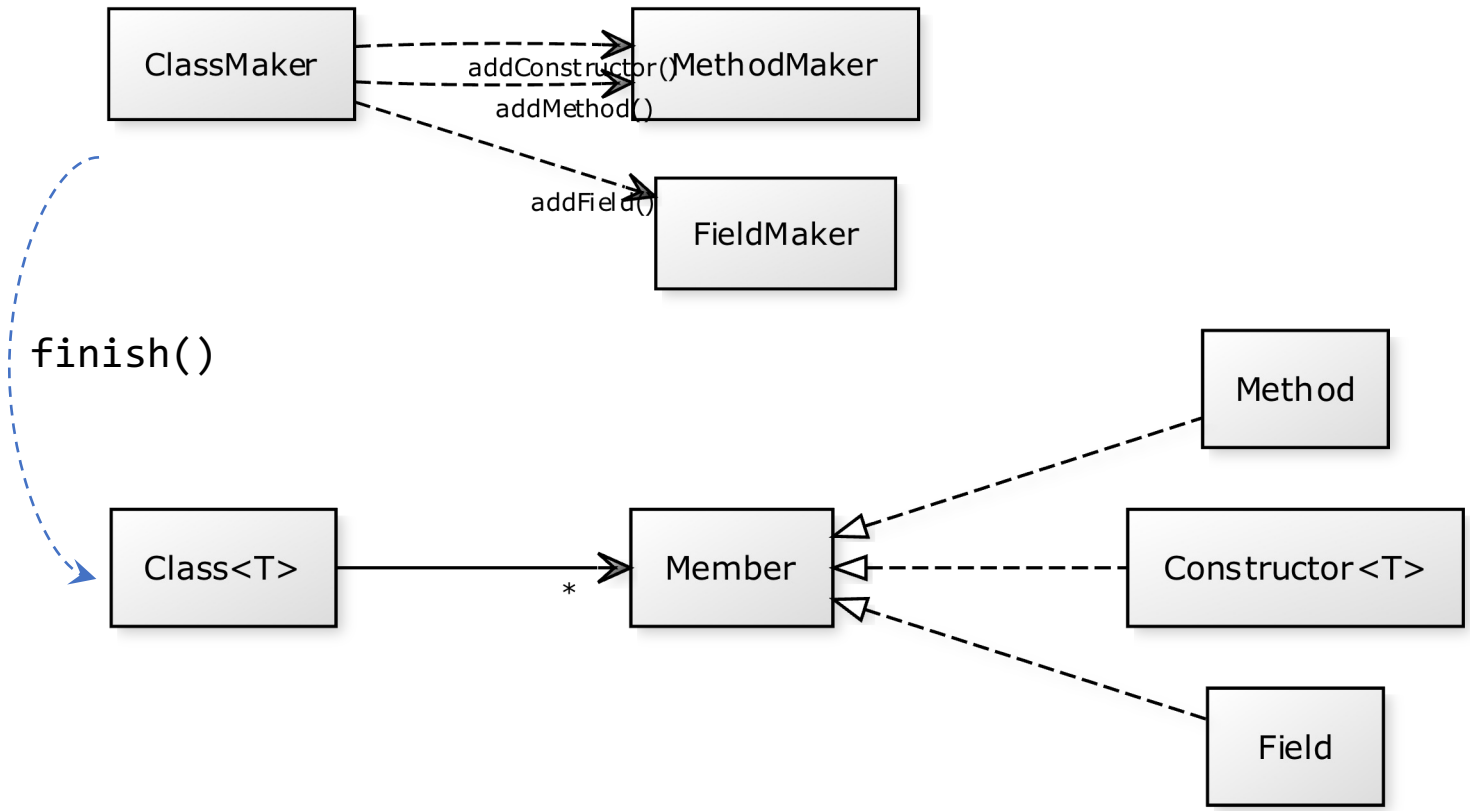
- .net - System.Reflection.Emit

- Java – third parties, e.g. *Javassist* (Java **bytecode** engineering toolkit), *ASM* (Java **bytecode** manipulation and analysis framework), *JavaPoet*, *Cojen Maker*, etc

- Javascript

  e.g. `foo.hello = function() { console.log('hello') }`

Cojen Maker


ClassMaker
addConstructor()
addMethod()
MethodMaker
addField()
FieldMaker
finish()
Class<T>
*
Member
Method
Constructor<T>
Field

Example

```
public final class MyDynamic {
  final int nr;

  public MyDynamic(int nr) {
    this.nr = nr;
  }
  public int mul(int other) {
    return this.nr * other;
  }
}
```

```
FieldMaker nrMaker = classMaker
        .addField(int.class, "nr")
        .private_()
        .final_();
```

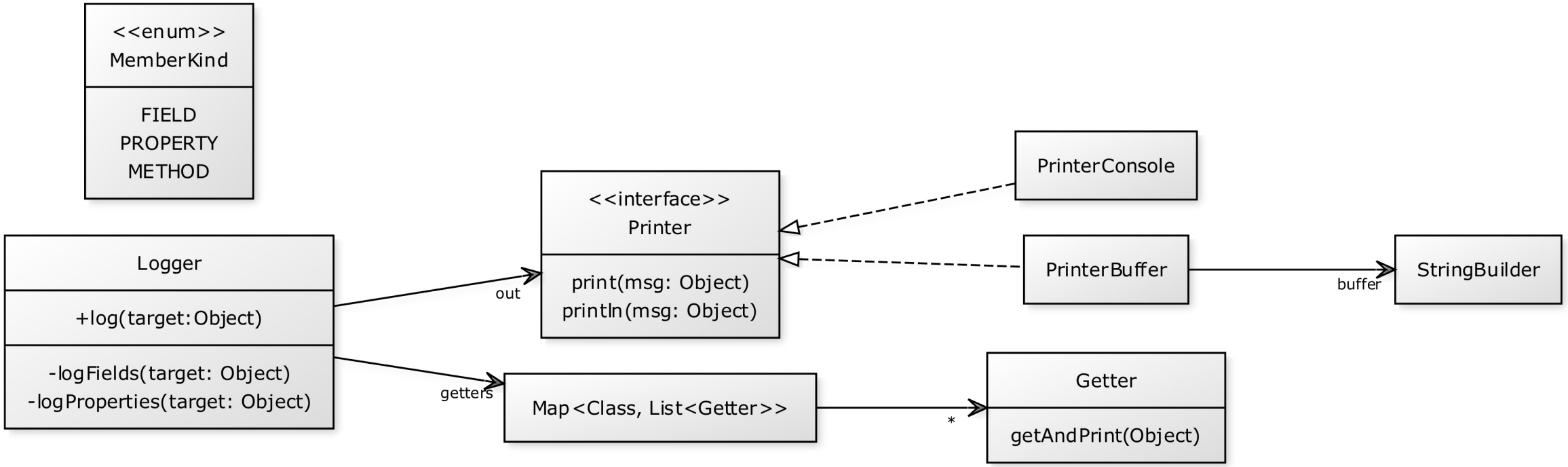
```
MethodMaker ctorMaker = classMaker
        .addConstructor(int.class)
        .public_();
ctorMaker
        .invokeSuperConstructor();
ctorMaker
        .field(nrMaker.name())
        .set(ctorMaker.param(0));
```

```
MethodMaker mulMaker = classMaker
        .addMethod(int.class, "mul", int.class)
        .public_();
Variable res = mulMaker
        .param(0)
        .mul(mulMaker.field(nrMaker.name()));
mulMaker.return_(res);
```

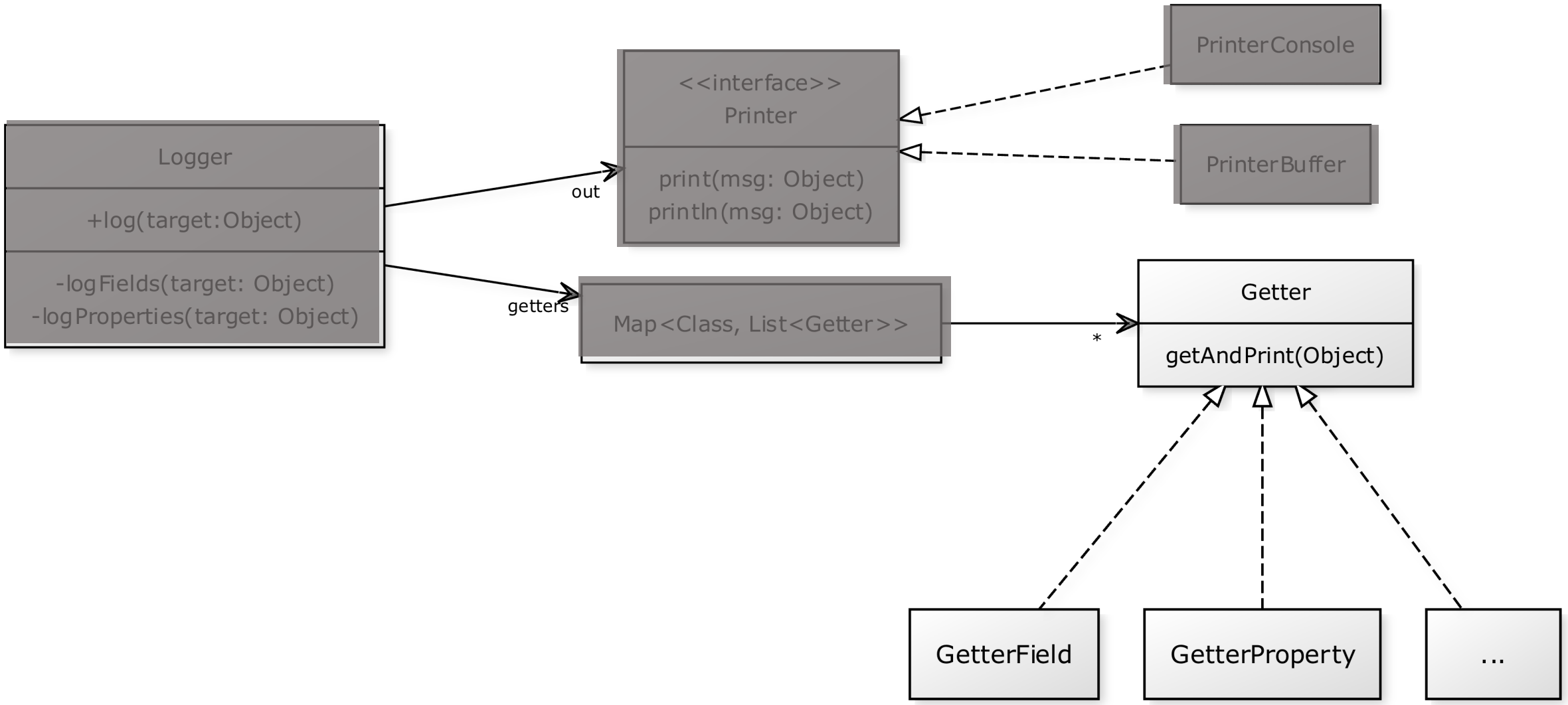
Outline

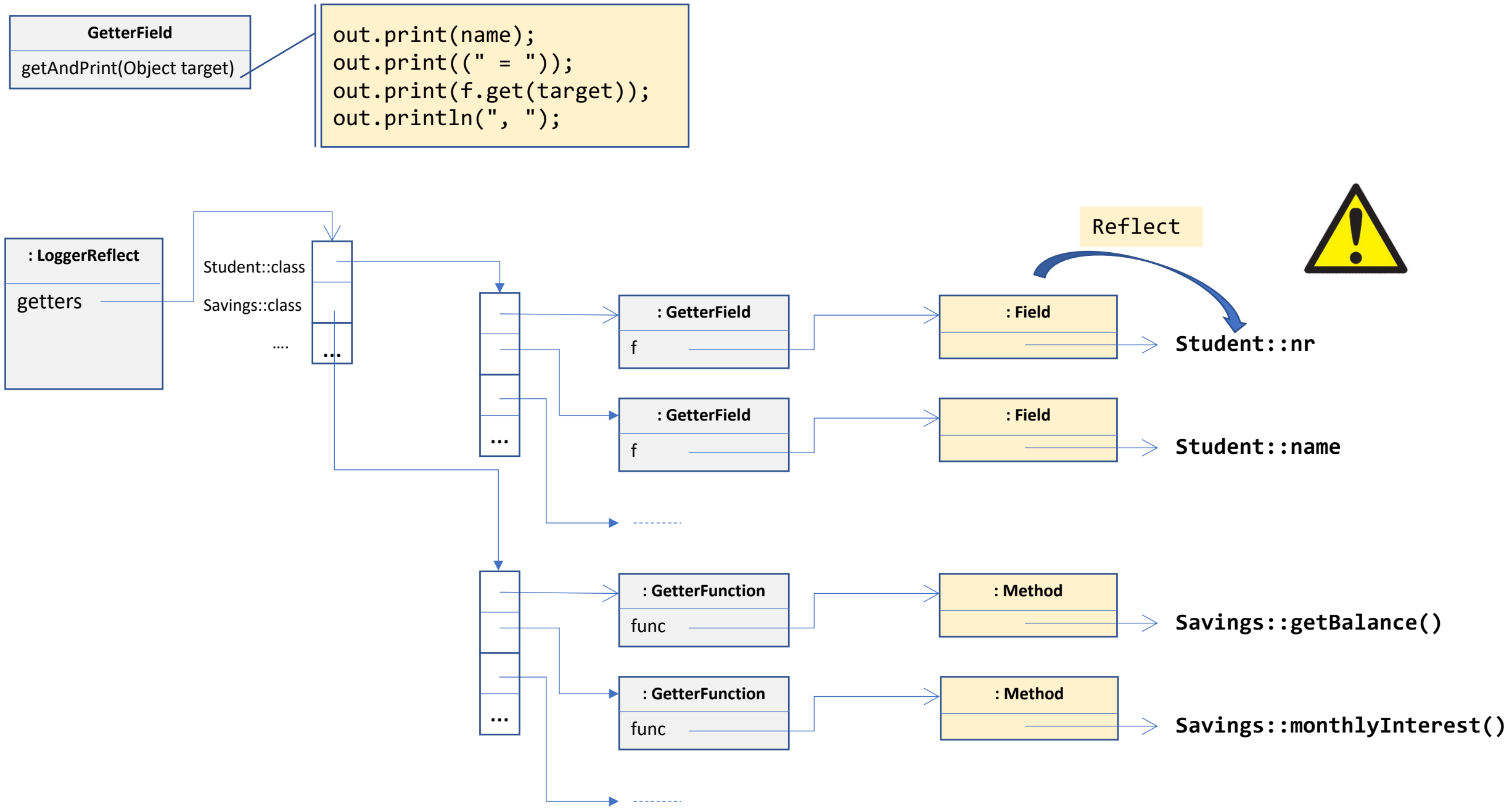
- Remember – Reflection
- Meta-programming e.g. .net Emit
- Remember – Logger via Reflect
- LoggerDynamic

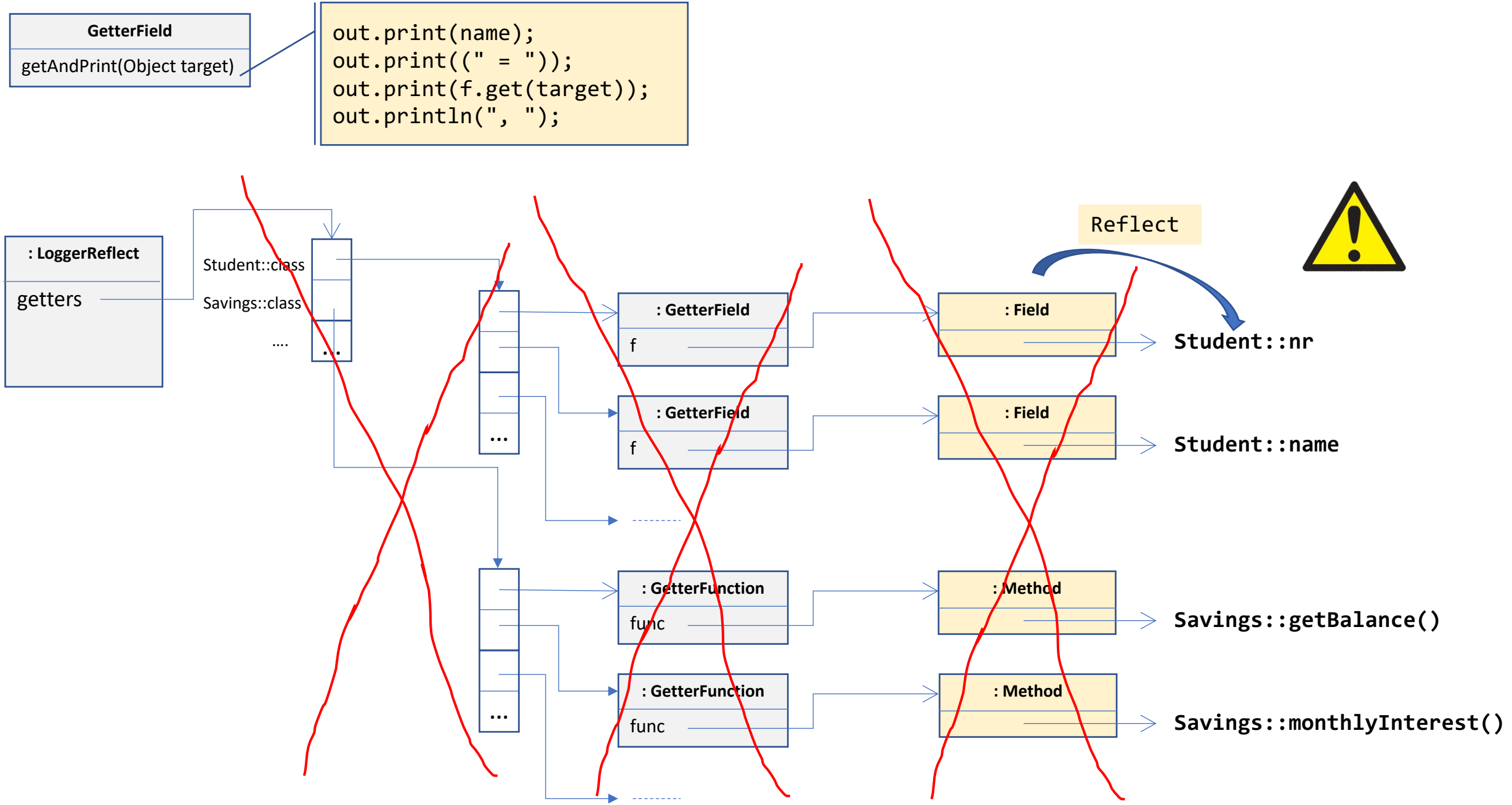
Remember – Logger via Reflect



Remember – Logger via Reflect




GetterField
getAndPrint(Object target)
out.print(name);
out.print((" = "));
out.print(f.get(target));
out.println(", ");
: LoggerReflect
getters
Student::class
Savings::class
....
...
: GetterField
f
: GetterField
f
: Field
: Field
Reflect
Student::nr
Student::name
: GetterFunction
func
: GetterFunction
func
: Method
: Method
Savings::getBalance()
Savings::monthlyInterest()


GetterField
getAndPrint(Object target)
out.print(name);
out.print((" = "));
out.print(f.get(target));
out.println(", ");
: LoggerReflect
getters
Student::class
Savings::class
....
: GetterField
f
: GetterField
f
: GetterFunction
func
: GetterFunction
func
: Field
: Field
: Method
: Method
Reflect
Student::nr
Student::name
Savings::getBalance()
Savings::monthlyInterest()

Outline

- Remember – Reflection
- Meta-programming e.g. .net Emit
- Remember – Logger
- LoggerDynamic


LoggerDynamic
logger(Class<?> domain)
: LoggerDynamicStudent
log(Object target)
: LoggerDynamicSavings
log(Object target)
: ...
log(Object target)
ALOAD 1
CHECKCAST pt/isel/Student
ASTORE 2
ALOAD 0
GETFIELD pt/isel/AbstractLogger.out : Lpt/isel/Printer;
LDC "Student: "
INVOKEINTERFACE pt/isel/Printer.print (Ljava/lang/Object;)V (itf)
ALOAD 0
GETFIELD pt/isel/AbstractLogger.out : Lpt/isel/Printer;
LDC "name = "
INVOKEINTERFACE pt/isel/Printer.print (Ljava/lang/Object;)V (itf)
ALOAD 2
GETFIELD pt/isel/Student.name : Ljava/lang/String;
ASTORE 3
ALOAD 0
GETFIELD pt/isel/AbstractLogger.out : Lpt/isel/Printer;
ALOAD 3
INVOKEINTERFACE pt/isel/Printer.print (Ljava/lang/Object;)V (itf)
ALOAD 0
GETFIELD pt/isel/AbstractLogger.out : Lpt/isel/Printer;
LDC ", "
INVOKEINTERFACE pt/isel/Printer.print (Ljava/lang/Object;)V (itf)
...
Student::nr
Student::name
...
LDC "balance = "
INVOKEINTERFACE pt/isel/Printer.print (Ljava/lang/Object;)V (itf)
ALOAD 2
INVOKEVIRTUAL pt/isel/Student.getBalance ()I
ASTORE 3
...
Savings::getBalance()
Savings::monthlyInterest()