

6. Java_MIDDLE (разбор рефлексии)

- Получение классов, методов, полей и аннотаций
- Динамические вызовы классов, методов, конструкторов
- Метод `invoke()`

Class Example: обращение и получение доступа вспомогательному классу `Person`, его полям и методам.

Для реализации метода `invoke()` используется класс `Scanner`, который считывает с клавиатуры 3 параметра: 1) название класса для которого будет вызван метод, 2) название класса, используемого в качестве параметра, 3) название метода

Пример : 1) `Thread` 2) `java.lang.String` 3) `setName`

Вывод: будет создан экземпляр класса `Thread`, с помощью метода `newInstance()`. И для него вызван метод `setName`, в качестве параметра использован конструктор (`getConstructor()`) из класса `java.lang.String`, и передано значение “String value”.

```
public class Example {
    public static void main(String[] args) throws ClassNotFoundException,
NoSuchMethodException, IllegalAccessException, InstantiationException,
InvocationTargetException {
        Person person = new Person();
        Class personClass = Person.class;           //1) Обратиться к классу
(название_класса.класс)
        Class personClass2 = person.getClass(); //2) Обратиться к экземпляру
класса (объект.getClass)
        Class personClass3 = Class.forName("Reflections.Person"); //3)
Обратиться по полному названию класса (forName(Указать путь))

        Method[] methods = personClass.getMethods();
        for (Method method : methods) {
            System.out.println(method.getName() + " , " +
method.getReturnType() + " , " +
Arrays.toString(method.getParameterTypes()));
        }

        Field[] fields = personClass.getDeclaredFields();
        for (Field field : fields) {
            System.out.println(field.getName() + " , " + field.getType());
        }

        Annotation[] annotations = personClass.getAnnotations();
        for (Annotation annotation : annotations) {
            System.out.println(annotation.toString());
        }
    }
}
```

```
Scanner sn = new Scanner(System.in);
Class classObj1 = Class.forName(sn.next()); //Название класса 1,
Class classObj2 = Class.forName(sn.next()); // Название класса 2
String methodName = sn.next(); //Название метода

Method m = classObj1.getMethod(methodName, classObj2);
Object o1 = classObj1.newInstance();
Object o2 =
classObj2.getConstructor(String.class).newInstance("String Value");
m.invoke(o1, o2);
System.out.println(o1);
}
```