```
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import java.lang.reflect.Type;

public class HeaterMain
{
	public static void main(String[] args)
	{
		//Bez refleksji:

		Heater heater=new Heater();
		heater.increment();
		System.out.println(heater.getTemperature());


		//Z refleksją:

		Class cl=null;
		Method m1=null, m2=null;

		try
		{
			cl=Class.forName("Heater");
		}
		catch(ClassNotFoundException e)
		{}

		try
		{
			m1=cl.getMethod("increment", null);
			m2=cl.getMethod("getTemperature", null);
		}
		catch(NoSuchMethodException e)
		{}
		catch(SecurityException e)
		{}

		try
		{
			Heater h=(Heater)cl.newInstance();
			m1.invoke(h, null);
			System.out.println(m2.invoke(h, null));
		}
		catch(IllegalAccessException e)
		{}
		catch(IllegalArgumentException e)
		{}
		catch(InvocationTargetException e)
		{}
		catch(InstantiationException e)
		{}

		System.out.println("Klasa "+cl.getName()+" ma metody:");
		Method[] methods =cl.getDeclaredMethods();

		for(int i=0; i<methods.length; ++i)
		{
			System.out.println(" Metoda "+methods[i].getName().toString());
			System.out.println("   która posiada parametry typu:  ");
			Type[] types=methods[i].getParameterTypes();

			for(int j=0; j<types.length; ++j)
			{
				System.out.println("      "+types[j].toString());
			}
			System.out.println("   która zwraca wartość typu:"
				+methods[i].getReturnType().toString());
		}
	}
}
```