```
package cn.trasen

import java.io.PrintWriter
import java.io.StringWriter
import java.lang.Exception

open class TException(var code:Int=999, var type:String="原生异常", var subtype:String="未知错误", var detail:String="发生了未知的
异常", cause:Throwable?=null):Throwable(message = "[${code}${type}] ${subtype}: $detail", cause = cause){

    var trace: String = ""

    companion object{
        fun dumpTrace(e: Throwable):String{
            val sw = StringWriter()
            val pw = PrintWriter(sw)
            e.printStackTrace(pw)
            return sw.toString()
        }
        fun findRootCause(throwable: Throwable):Throwable {
            if (throwable.cause == null) return throwable
            return findRootCause(throwable.cause!!)
        }
        private fun getThrowableMessage(throwable: Throwable): String {
            return when (throwable) {
                is TException -> throwable.message!!
                else -> "[原生异常] ${throwable.javaClass}"
            }
        }
    }

    fun initInfo(){
        val rootCause = findRootCause()
        if(rootCause is TException){
            code = rootCause.code
            type = rootCause.type
            subtype = rootCause.subtype
            detail = rootCause.detail
        }
        trace = dumpTrace()
    }

    private fun findRootCause():Throwable{
        return findRootCause(this)
    }

    private fun dumpTrace():String{
        return dumpTrace(this)
    }
}

// 通用异常
open class CommonException(code:Int, type:String, subtype:String, detail: String, cause:Throwable?):
    TException(code=1000+code, type="通用异常", subtype=subtype, detail=detail, cause=cause)
// 编译异常
open class CompilerException(code:Int, subtype:String, detail: String, cause:Throwable?):
    TException(code=2000+code, type="编译异常", subtype=subtype, detail=detail, cause=cause)
// 反编译异常
open class DecompilerException(code:Int, subtype:String, detail: String, cause:Throwable?):
    TException(code=3000+code, type="反编译异常", subtype=subtype, detail=detail, cause=cause)

// 这是都是编译器的异常
open class CompilerParsingException(code:Int, subtype:String, msg:String, cause:Throwable?):
```

```
    CompilerException(code=100+code, subtype="解析器异常-${subtype}", detail=msg, cause=cause)
open class CompilerWalkingException(code:Int, subtype:String, msg:String, cause:Throwable?):
    CompilerException(code=200+code, subtype="遍历器异常-${subtype}", detail=msg, cause=cause)
open class CompilerVisitingException(code:Int, subtype:String, msg:String, cause:Throwable?):
    CompilerException(code=300+code, subtype="访问器异常-${subtype}", detail=msg, cause=cause)
open class CompilerRenderingException(code:Int, subtype:String, msg:String, cause:Throwable?):
    CompilerException(code=400+code, subtype="渲染器异常-${subtype}", detail=msg, cause=cause)

//未知异常
open class UnknownException: TException {
    constructor(cause:Throwable?=null):super(0, "未知异常", "未知异常", "发生了非预期的异常", cause)
    constructor(msg: String, cause:Throwable?=null):super(0,"未知异常", "未知异常", msg,cause)
}
class CompilerUnknownException(msg: String, cause:Throwable?=null): CompilerException(0, "未知编译异常", msg, cause)
class DecompilerUnknownException(msg: String, cause:Throwable?=null): DecompilerException(0, "未知反编译异常", msg, cause)


// 解析错误
class CompilerParsingUnknownException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerParsingException(0, "未知解析异常", msg, cause)
class CompilerParsingGraphException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerParsingException(1, "图结构解析异常", msg, cause)
class CompilerParsingEdgesException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerParsingException(2, "边集合解析异常", msg, cause)
class CompilerParsingNodesException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerParsingException(3, "点集合解析异常", msg, cause)
class CompilerParsingEdgeException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerParsingException(4, "结点解析异常", msg, cause)
class CompilerParsingNodeException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerParsingException(5, "边解析异常", msg, cause)
class CompilerParsingExpressionException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerParsingException(6, "表达式解析异常", msg, cause)

//遍历错误
class CompilerWalkingUnknownException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerWalkingException(0, "未知遍历异常", msg, cause)
class CompilerWalkingGraphException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerWalkingException(1, "图遍历异常", msg, cause)
class CompilerWalkingEdgesException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerWalkingException(2, "边遍历异常", msg, cause)
class CompilerWalkingNodesException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerWalkingException(3, "节点遍历异常", msg, cause)
class CompilerWalkingExpressionException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerWalkingException(4, "表达式遍历异常", msg, cause)

//访问错误
class CompilerVisitingUnknownException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerVisitingException(0, "未知解析异常", msg, cause)
class CompilerVisitingGraphException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerVisitingException(1, "图解析异常", msg, cause)
class CompilerVisitingEdgesException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerVisitingException(2, "边解析异常", msg, cause)
class CompilerVisitingNodesException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerVisitingException(3, "节点解析异常", msg, cause)
class CompilerVisitingExpressionException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerVisitingException(4, "表达式解析异常", msg, cause)
```

```
//渲染错误
class CompilerRenderingUnknownException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerRenderingException(0, "未知渲染异常", msg, cause)
class CompilerRenderingGraphException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerRenderingException(1, "图渲染异常", msg, cause)
class CompilerRenderingEdgesException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerRenderingException(2, "边渲染异常", msg, cause)
class CompilerRenderingNodesException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerRenderingException(3, "节点渲染异常", msg, cause)
class CompilerRenderingExpressionException(msg:String="暂无描述", cause:Throwable?=null):
    CompilerRenderingException(4, "表达式渲染异常", msg, cause)



inline fun <reified T> tryAndCatch(exc:(msg:String, e:Throwable)-> TException, func:()-> T): T {
    try{
        return func()
    }catch(e: Exception){
        val error = exc(e.message?:"未知错误", e)
        error.initInfo()
        throw error
    }
}
```