/\*\*

\* We declare a package-level function main which returns Unit and takes

\* an Array of strings as a parameter. Note that semicolons are optional.

\*/

open class Report(val title: String="",

val imageURL: String="",

val description: String=""){

open fun getReportDetails(): String {

return title + ": " + description

}

}

//create the abstract class for the decorator

abstract class ReportDecorator(val report: Report) : Report(report.title, report.imageURL, report.description){

override abstract fun getReportDetails(): String

}

//create the location decorator

class LocationReportDecorator(report: Report, val longitude: Int, val latitude: Int) : ReportDecorator(report){

override fun getReportDetails(): String {

return report.getReportDetails() + " posted at [" + longitude + ", " + latitude + "]"

}

}

//create the urgent decorator

class UrgentReportDecorator(val status: String, report: Report) : ReportDecorator(report){

override fun getReportDetails(): String{

return status + " " + report.getReportDetails()

}

}

//create the day decorator

class DayReportDecorator(report: Report, val day: String) : ReportDecorator(report){

override fun getReportDetails(): String{

return report.getReportDetails() + " on " + day

}

}

fun main(args: Array<String>) {

//get the report

val report = Report("Report", "images/a.jpg", "description")

println(report.getReportDetails())

//try the location decorator

val locationReport = LocationReportDecorator(report, 60, 100)

println(locationReport.getReportDetails())

//try the urgent decorator

val urgentReport = UrgentReportDecorator("URGENT", report)

println(urgentReport.getReportDetails())

//try the day decorator

val dayReport = DayReportDecorator(report, "Wednesday")

println(dayReport.getReportDetails())

//try all the decorators

val newReport = DayReportDecorator(UrgentReportDecorator("URGENT", LocationReportDecorator(Report("Report", "images/a.jpg", "description"), 60, 100)), "Wednesday")

println(newReport.getReportDetails())

}