|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Strategy Pattern  Group 16 |
| |  |  |  | | --- | --- | --- | | Lyubomir Dimov , Nikola Nikushev | 2/21/16 | Design Patterns | |

Contents

[Introduction 2](#_Toc443832971)

[About Strategy Pattern 3](#_Toc443832972)

[Class Diagram 4](#_Toc443832973)

[Our Application and how it works 4](#_Toc443832974)

[Conclusion 4](#_Toc443832975)

# Introduction

This document gives explanations about Assignment 1 from the Design Pattern course. You will find explanation of what is Strategy Pattern, how people used it and in which situations is used.

To present how actually Strategy Pattern works, an application was implemented. The whole process of work was in the following sequence - research about the pattern, understanding the pattern, making proper class diagram of the application, implementation, testing, documentation.

The application is about Disc Scheduling of an Operating System, using three types of scheduling FCFS, SSTF, SCAN.

In this document you will find:

* Strategy Pattern – What is it, how it’s used and why.
* Class diagram of Disc Scheduling application with a brief explanation of it.
* Pictures and explanation of Disc Scheduling application.
* The graphical user interface and what is each button used for.
* How we implemented and tested the scheduling methods
* Multithreading in the application

Let’s first start with the concept of Strategy Pattern.

# About Strategy Pattern

## What is strategy?

Strategy is a plan of action to achieve a specific goal.

## When is used?

A situation in which you are given context in which an algorithm is used, and you want to use different algorithms depending upon the situation or to avoid having to change the context. In this kind of situations is preferably to use Strategy pattern.

## What is Strategy pattern?

“Define a family of algorithms, encapsulate each one, and make them interchangeable. Strategy lets the algorithm vary independently from clients that use it.” – (Internet)

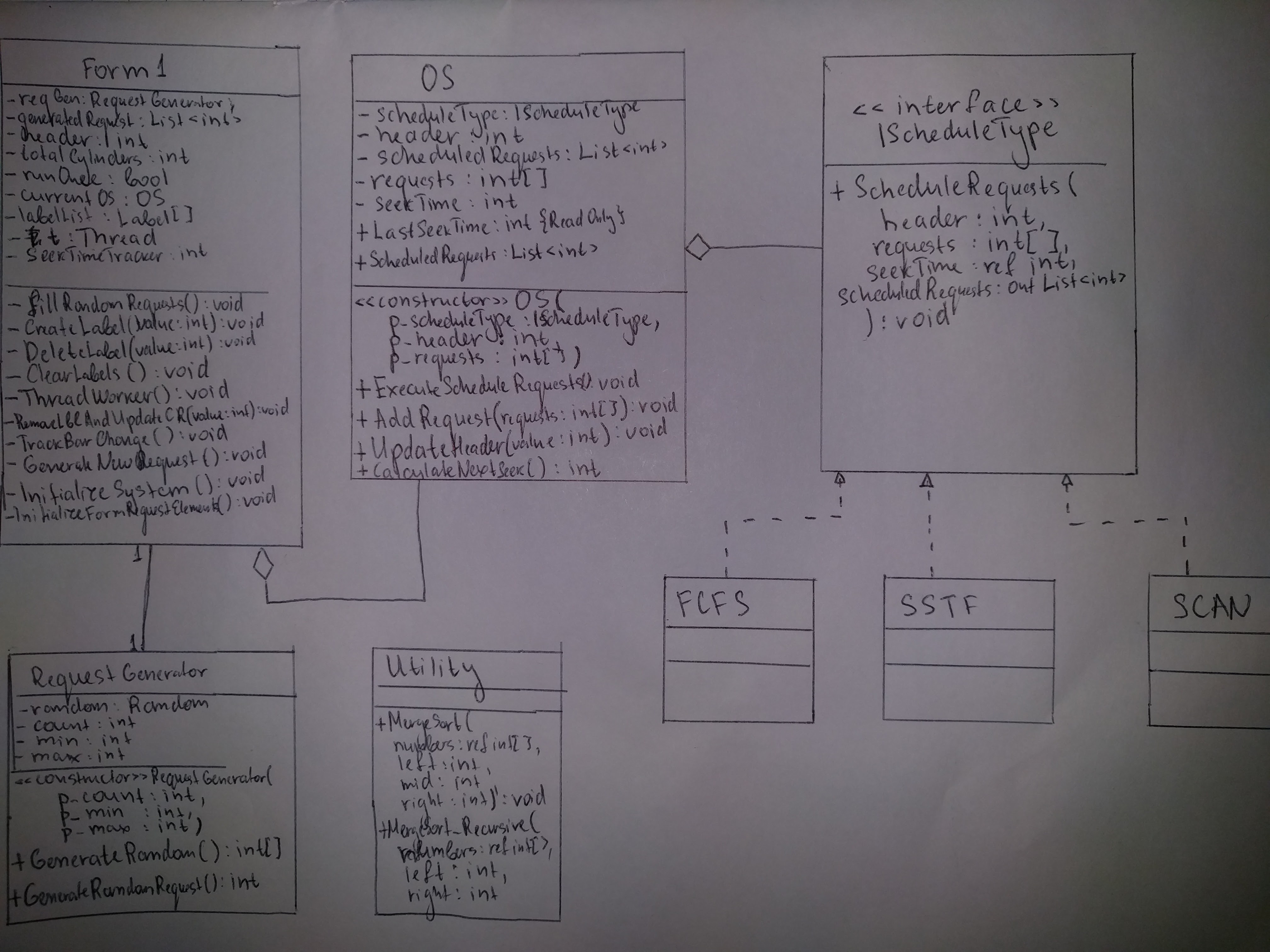


# Class Diagram

The class diagram was the initial part of the development of the application.

To apply properly the Strategy pattern we had to:

* Identify the aspects of our application that vary and separate them from what states the same.
* Program to an interface, not implementation
* Favor aggregation over inheritance



(note: there is an attached picture of the class diagram in the submission archive)

# Our Application and how it works

# Conclusion