객체 지향 프로그래밍 프로젝트 제안서: 태스크 스케줄러 시뮬레이터 🚀

### 1. 프로젝트 개요

이 프로젝트는 \*\*자바(Java)\*\* 언어를 사용하여 \*\*운영체제(OS)\*\*의 핵심 기능 중 하나인 \*\*태스크 스케줄링\*\*을 시뮬레이션하는 프로그램입니다. 일반적인 프로그램이 사용자의 직접적인 조작에 의존하는 것과 달리, 이 프로젝트는 사용자가 입력한 여러 작업(태스크)을 \*\*자동으로 처리하는 시스템의 동작 원리\*\*를 구현하는 데 중점을 둡니다.

\*\*궁극적인 목표:\*\* 사용자가 직접 코딩한 스케줄링 알고리즘에 따라 태스크의 실행 순서가 결정되는 과정을 시뮬레이션하여, 시스템의 논리적인 작동 방식을 깊이 있게 이해하는 것입니다. 이는 향후 \*\*임베디드 시스템\*\*과 같은 저수준(low-level) 개발 분야에 대한 좋은 경험이 될 것입니다.

---

### 2. 주요 기능 및 구현 방향

프로젝트는 크게 세 가지 핵심 기능으로 구성됩니다.

#### \*\*2.1. 태스크(Task) 관리\*\*

\* \*\*태스크 생성:\*\* 사용자가 태스크의 \*\*이름\*\*, \*\*실행 시간\*\*, \*\*우선순위\*\* 등의 속성을 입력하여 태스크를 생성할 수 있어야 합니다.

\* \*\*객체 모델링:\*\* 각 태스크는 \*\*`Task` 클래스\*\*로 모델링되며, 이 클래스는 태스크의 상태(준비, 실행, 완료)를 관리하는 메서드를 포함합니다.

\* \*\*데이터 구조:\*\* 생성된 태스크들을 관리하기 위해 \*\*큐(Queue)\*\*나 \*\*리스트(List)\*\*와 같은 적절한 \*\*자료구조\*\*를 활용합니다.

#### \*\*2.2. 스케줄러(Scheduler) 구현\*\*

\* \*\*알고리즘 구현:\*\* 태스크의 실행 순서를 결정하는 핵심 로직입니다. 처음에는 가장 간단한 \*\*`FIFO` (First-In, First-Out)\*\* 알고리즘을 구현하고, 여유가 있다면 \*\*`Round-Robin`\*\* 알고리즘을 추가 구현하여 두 알고리즘의 차이를 비교해볼 수 있습니다.

\* \*\*다형성 활용:\*\* `Scheduler`라는 \*\*인터페이스\*\*를 정의하고, `FIFOScheduler`와 `RoundRobinScheduler`가 이를 \*\*구현\*\*하도록 설계하여 \*\*다형성\*\*을 적용합니다. 이를 통해 스케줄링 알고리즘을 쉽게 교체하고 확장할 수 있습니다.

#### \*\*2.3. 시뮬레이션 엔진\*\*

\* \*\*실행 과정 출력:\*\* 스케줄러가 결정한 순서에 따라 태스크가 실행되는 과정을 시간 흐름에 맞춰 텍스트로 출력합니다. 예를 들어, "Time: 1s, Task A 실행 중", "Time: 3s, Task B 실행 완료" 와 같은 형태로 보여줍니다.

\* \*\*로그 기록:\*\* 각 태스크의 시작, 완료 시점과 같은 중요한 이벤트를 기록하여, 시뮬레이션이 끝난 후 전체 실행 내역을 한눈에 볼 수 있도록 합니다.

---

### 3. 기술 스택 및 개발 환경

\* \*\*주요 언어:\*\* Java

\* \*\*개발 환경:\*\* 이클립스(Eclipse), 인텔리제이(IntelliJ) 등 원하는 IDE

\* \*\*구현 범위:\*\* 높은 수준의 그래픽 UI는 필요하지 않습니다. 모든 출력은 콘솔(터미널)을 통해 진행됩니다.

---

### 4. 예상 효과 및 기대 결과

\* \*\*객체 지향 원리 숙달:\*\* \*\*클래스, 객체, 상속, 다형성, 인터페이스\*\* 등 핵심 객체 지향 개념을 프로젝트에 직접 적용하며 몸으로 익힐 수 있습니다.

\* \*\*시스템 사고 능력 향상:\*\* 단순한 데이터 관리를 넘어, \*\*시스템의 동작 로직\*\*을 설계하고 구현하는 경험을 통해 문제 해결 능력을 키울 수 있습니다.

\* \*\*임베디드 개발 기초 다지기:\*\* \*\*태스크 스케줄링\*\*은 임베디드 시스템 개발에서 매우 중요한 개념입니다. 이 프로젝트는 관련 지식을 간접적으로나마 경험해볼 수 있는 좋은 토대가 될 것입니다.

---

### 5. 진행 로드맵 (2개월 기준)

\* \*\*1개월차:\*\*

\* \*\*1주차:\*\* 프로젝트 설계 (클래스 다이어그램 작성), `Task` 클래스 구현

\* \*\*2-3주차:\*\* `FIFOScheduler` 구현, 간단한 시뮬레이션 엔진 개발

\* \*\*4주차:\*\* 첫 번째 구현 버전 테스트 및 코드 리팩토링

\* \*\*2개월차:\*\*

\* \*\*5-6주차:\*\* `RoundRobinScheduler` 구현, 스케줄링 알고리즘 선택 기능 추가

\* \*\*7주차:\*\* 시뮬레이션 결과에 대한 통계 정보(평균 대기 시간 등) 추가

\* \*\*8주차:\*\* 최종 코드 정리 및 문서화

이 프로젝트는 단순히 코드를 작성하는 것을 넘어, 컴퓨터 시스템이 어떻게 작동하는지에 대한 근본적인 질문에 답을 찾는 과정이 될 것입니다.