**“Page Replacement Simulator”**

**REQUIREMENTS**

* 다양한 페이지 교체 정책 시뮬레이션 제공
* 데이터 스트림을 직접 입력하거나 파일로 입력
* Hit, Miss, Hit Ratio 계산
* 페이지 교체 애니메이션 출력
* 결과 보고서 출력, 결과 파일 생성

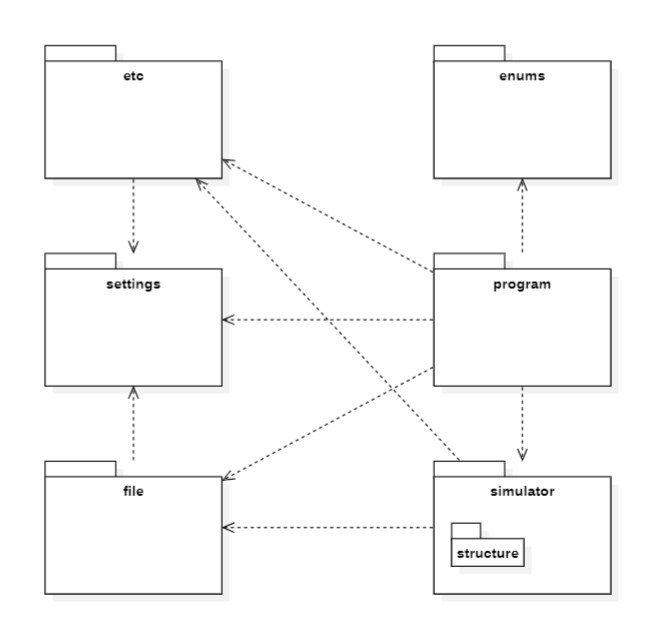
**DEVELOPMENT**

* Java
* IntelliJ

**LIBRARY**

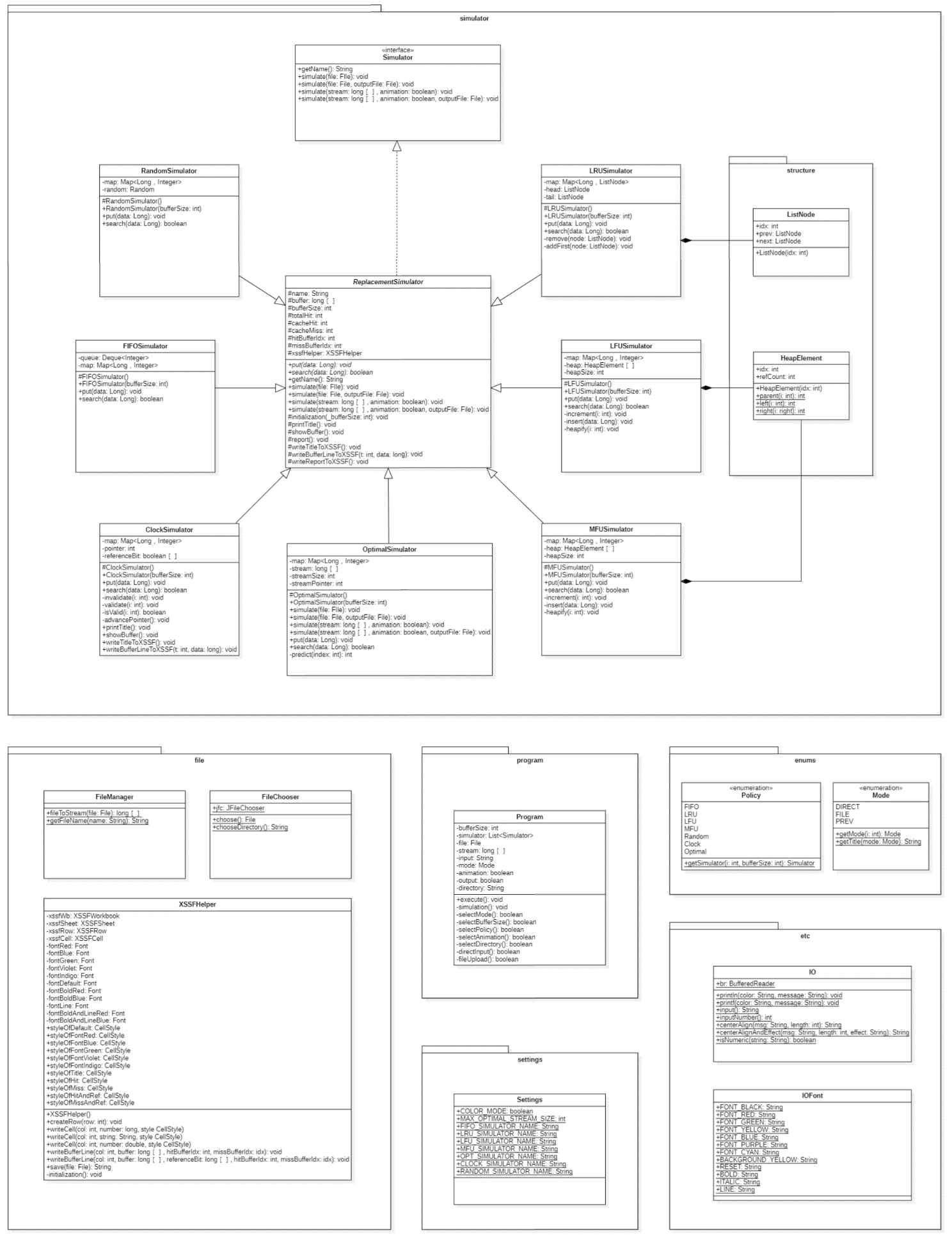
* poi-bin-5.2.3
  + <https://www.apache.org/dyn/closer.lua/poi/release/bin/poi-bin-5.2.3-20220909.zip>
* log4j-core-2.19.0
  + <https://www.apache.org/dyn/closer.lua/logging/log4j/2.19.0/apache-log4j-2.19.0-bin.zip>

**PACKAGES**



|  |  |
| --- | --- |
| **패키지** | **설명** |
| program | Simulator를 동작시키는 Program Class을 제공하는 패키지 |
| simulator | Simulator를 제공하는 패키지 |
| structure | Simulator 구현에 필요한 자료구조 Class를 제공하는 패키지 |
| file | 파일 선택 다이얼로그, 엑셀 저장 라이브러리 Class를 제공하는 패키지 |
| settings | 프로그램 설정 정보 Class를 제공하는 패키지 |
| etc | IO에 관련된 Class를 제공하는 패키지 |
| enums | 프로젝트에 관련된 Enum Class를 제공하는 패키지 |

**CLASS DIAGRAM**



**CLASS**

|  |  |  |  |
| --- | --- | --- | --- |
| **패키지** | **종류** | **이름** | **기능** |
| enums | E | Modes | • 데이터 입력 모드 Enumeration |
| enums | E | Policy | • 페이지 교체 정책 Enumeration  • Simulator 반환하는 역할 |
| etc | C | IO | • 출력문 가운데 정렬, 입력 기능  • 출력문 색상, 효과 설정 |
| etc | C | IOFont | • IO Class에서 사용하는 폰트, 효과 |
| file | C | FileChooser | • 파일 선택 다이얼로그 기능  • 저장할 폴더 선택 기능 |
| file | C | FileManager | • 파일을 스트림 배열로 변환(OPT Simulator)  • 저장할 파일 이름 생성 |
| file | C | XSSFHelper | • 엑셀 저장 외부 라이브러리 사용  • 엑셀 파일 스타일 정의  • 엑셀 파일 쓰기, 저장 |
| program | C | Program | • 사용자로부터 정보를 입력 받음  • 시뮬레이터 구현, 동작 |
| settings | C | Settings | • 설정 |
| simulator | C | ClockSimulator | • Clock Replacement (Second Chance Replacement)  • 원형 큐 |
| simulator | C | FIFOSimulator | • First-In First-Out Replacement |
| simulator | C | LFUSimulator | • Least Frequently Used Replacement  • 최소 힙 |
| simulator | C | LRUSimulator | • Least Recently Used Replacement  • 이중연결리스트 |
| simulator | C | MFUSimulator | • Most Frequently Used Replacement  • 최대 힙 |
| simulator | C | OptimalSimulator | • Optimal Replacement  • O(n2) |
| simulator | C | RandomSimulator | • Random Replacement |
| simulator | (C) | ReplacementSimulator | • Simulator 인터페이스 구현 |
| simulator | I | Simulator | • Simulator 인터페이스  • 시뮬레이터를 실행하는 simulate 제공 |
| simulator.  structure | C | HeapElement | • LFU, MFU 힙을 위한 원소 |
| simulator.  structure | C | ListNode | • LRU 이중연결리스트를 위한 노드 |
| - | C | Main | • Program 실행 |

**E: Enum I: 인터페이스 C: 클래스 (C): 추상클래스 .**

**SIMULATOR CLASSES DETAIL**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Simulator 인터페이스   |  |  | | --- | --- | | **메소드** | **설명** | | getName | • 시뮬레이터 이름 반환 | | simulate | • 입력 데이터를 사용하여 페이지 교체 시뮬레이션 동작  • 4가지 오버로딩  - 파일로 데이터 입력  - 파일로 데이터 입력, 결과 파일 저장  - 배열로 데이터 입력  - 배열로 데이터 입력, 결과 파일 저장 | |
|  | | Simulator 구현체, 모든 시뮬레이터가 가지는 공통 기능을 가진 클래스   |  |  | | --- | --- | | **필드** | **설명** | | name | • 시뮬레이터 이름 | | buffer | • 데이터가 저장되는 공간 | | bufferSize | • buffer의 최대 크기 | | totalHit | • 입력된 데이터의 총 개수 | | cacheHit | • 캐시 히트를 세는 카운터 | | cacheMiss | • 캐시 미스(페이지 폴트)를 세는 카운터 | | hitBufferIdx | • hit가 일어난 buffer의 인덱스 저장 | | missBufferIdx | • miss가 일어난 buffer의 인덱스 저장 | | xssfHelper | • 엑셀 파일로 저장하기 위한 XSSFHelper |  |  |  | | --- | --- | | **메소드** | **설명** | | put | • 데이터를 교체 정책에 따라 buffer에 입력  • 교체 정책마다 다르게 구현 | | search | • 입력된 데이터가 buffer에 있는지 판단 | | getName | • Simulator 인터페이스 구현 | | simulate | • Simulator 인터페이스 구현 | | initialization | • 필드 값 초기화 | | printTitle | • 애니메이션 결과 출력 시 상단 부분 문구 출력 | | showBuffer | • buffer 상태 출력 | | report | • 시뮬레이션의 결과 출력 | | writeTitleToXSSF | • XSSF 파일 셀에 상단 부분 입력 | | writeBufferLineToXSSF | • XSSF 파일 셀에 버퍼 내용 입력 | | writeReportToXSSF | • XSSF 파일 셀에 시뮬레이션 결과 입력 | |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | | FIFO Simulator   * 입력된 순서를 저장하고 있는 큐 자료구조로 구현  |  |  | | --- | --- | | **필드** | **설명** | | queue | • 입력된 순서를 유지하는 큐 자료구조 | | map | • buffer에 존재하는지 여부를 빠르게 확인하기 위한 맵 | |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | LRU Simulator   * 이중연결리스트로 구현 * 최근에 참조된 데이터일수록 head쪽 * PF 발생 시, 리스트 맨 뒤 노드를 제거하고 맨 앞에 새로운 노드 추가 * HIT 발생 시, 해당 노드를 맨 앞으로 이동  |  |  | | --- | --- | | **필드** | **설명** | | map | • buffer에 존재하는지 여부를 빠르게 확인하기 위한 맵 | | head | • double linked list를 위한 헤드 더미 노드 | | tail | • double linked list를 위한 테일 더미 노드 |  |  |  | | --- | --- | | **메소드** | **설명** | | remove | • double linked list에서 원소를 제거 | | addFirst | • double linked list의 맨 앞에 원소를 삽입 | | |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | LFU Simulator   * 최소 힙으로 구현  |  |  | | --- | --- | | **필드** | **설명** | | map | • buffer에 존재하는지 여부를 빠르게 확인하기 위한 맵 | | heap | • 힙 자료구조 | | heapSize | • 힙 크기 |  |  |  | | --- | --- | | **메소드** | **설명** | | increment | • 해당 인덱스의 원소 값을 1 증가시키고, 다시 힙 구성 | | insert | • 힙 삽입 연산 | | heapify | • heap 배열을 최소 힙 구조에 맞게 재배치 | | |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | MFU Simulator   * 최대 힙으로 구현 * LFU Simulator와 동일 | |
|  | Clock Simulator   * 원형 큐로 구현 * 현재 포인터의 reference bit가 true라면 false로 만들고 전진 * 현재 포인터의 reference bit가 false일 때, 해당 인덱스의 프레임 교체 * 다른 시뮬레이터와 다르게 애니메이션에서 pointer 위치도 출력 * printTitle(), showBuffer() 메소드 오버라이딩  |  |  | | --- | --- | | **필드** | **설명** | | map | • buffer에 존재하는지 여부를 빠르게 확인하기 위한 맵 | | pointer | • Clock Pointer | | referenceBit | • buffer의 reference 상태를 저장하는 배열 |  |  |  | | --- | --- | | **메소드** | **설명** | | invalidate | • referenceBit 배열 해당 인덱스의 값을 false로 설정 | | validate | • referenceBit 배열 해당 인덱스의 값을 true로 설정 | | isValid | • referenceBit 배열 해당 인덱스의 값을 반환 | | advancdPointer | • pointer 필드의 값을 1 증가, buffer 크기가 되면 0 | | printTitle | • 다른 시뮬레이터와 출력문이 다르기 때문에 오버라이딩 | | showBuffer | • 다른 시뮬레이터와 출력문이 다르기 때문에 오버라이딩 | | writeTitleToXSSF | • 다른 시뮬레이터와 출력문이 다르기 때문에 오버라이딩 | | writeBufferLineToXSSF | • 다른 시뮬레이터와 출력문이 다르기 때문에 오버라이딩 | | |
|  | Optimal Simulator   * 매번 입력될 때 마다 predict 메소드로 교체할 페이지 선택 * predict 메소드가 O(n2)이므로 스트림 길이가 너무 크면 오래 걸림 * Settings에 동작할 최대 길이 설정  |  |  | | --- | --- | | **필드** | **설명** | | map | • buffer에 존재하는지 여부를 빠르게 확인하기 위한 맵 | | stream | • 입력된 데이터 스트림을 한 번에 저장하고 있는 배열 | | streamSize | • stream 배열의 크기 | | streamPointer | • 현재 처리중인 stream 원소의 인덱스를 가리키는 포인터 |  |  |  | | --- | --- | | **메소드** | **설명** | | simulate | • 다른 시뮬레이터와 동작 방식이 다르기 때문에 오버라이딩 | | predict | • 앞으로 가장 오랫동안 사용되지 않을 buffer의 인덱스를 반환 | | |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | Random Simulator   * 무작위로 buffer에 접근하여 해당 페이지 교체  |  |  | | --- | --- | | **필드** | **설명** | | map | 데이터가 buffer에 존재하는지 여부를 빠르게 확인하기 위한 맵 | | random | buffer에 접근할 무작위 인덱스를 생성하기 위한 필드 | | |

**DESIGN PATTERN**

* Template Method
  + simulate() 메서드
  + 추상클래스 ReplacementSimulator에 시뮬레이션을 위한 메서드를 protected로 선언
  + 각 페이지 교체 정책 시뮬레이터마다 ReplacementSimulator를 상속받아 시뮬레이션을 위한 메서드를 교체 정책에 맞게 구현

\* abstract void put(Long data);

\* abstract boolean search(Long data);

* + 각 페이지 교체 정책 시뮬레이터는 교체 정책에 맞는 put, search 메서드를 구현
  + 필요에 따라 다른 메서드도 오버라이딩
* Strategy
  + 사용자가 교체 정책을 사용할 때마다 전략을 변경할 수 있도록 사용
  + 클라이언트인 Policy.java 클래스가 구체적인 전략을 알아야 함

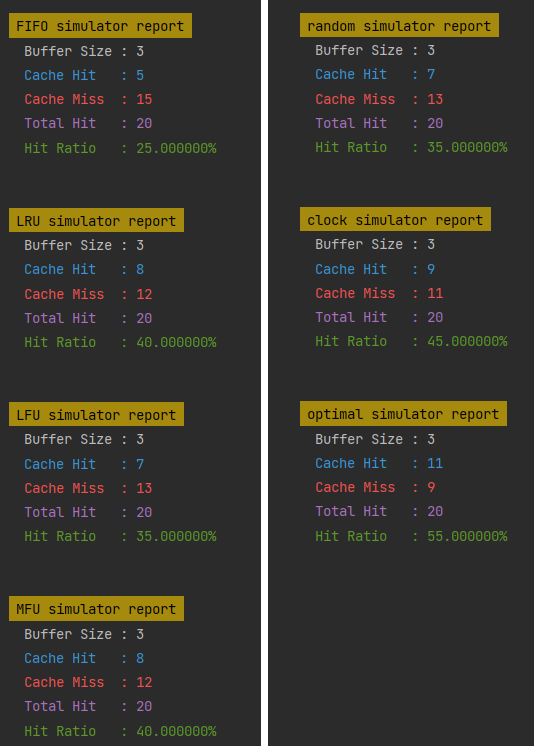
**PRODUCT**

입력 예시

텍스트이(가) 표시된 사진

자동 생성된 설명

결과 출력



테이블이(가) 표시된 사진

자동 생성된 설명애니메이션 출력(예시: optimal simulator)

엑셀 저장(예시: clock simulator)

테이블이(가) 표시된 사진

자동 생성된 설명 