“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

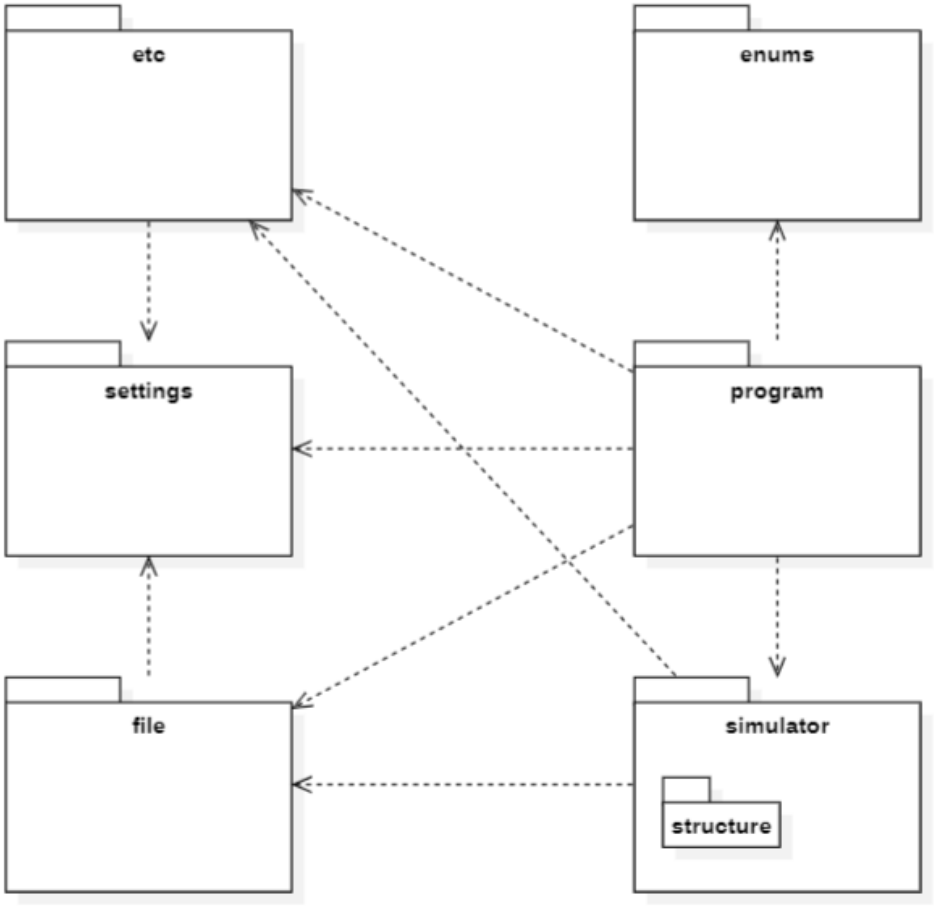
DEVELOPMENT

- Java
- IntelliJ

LIBRARY

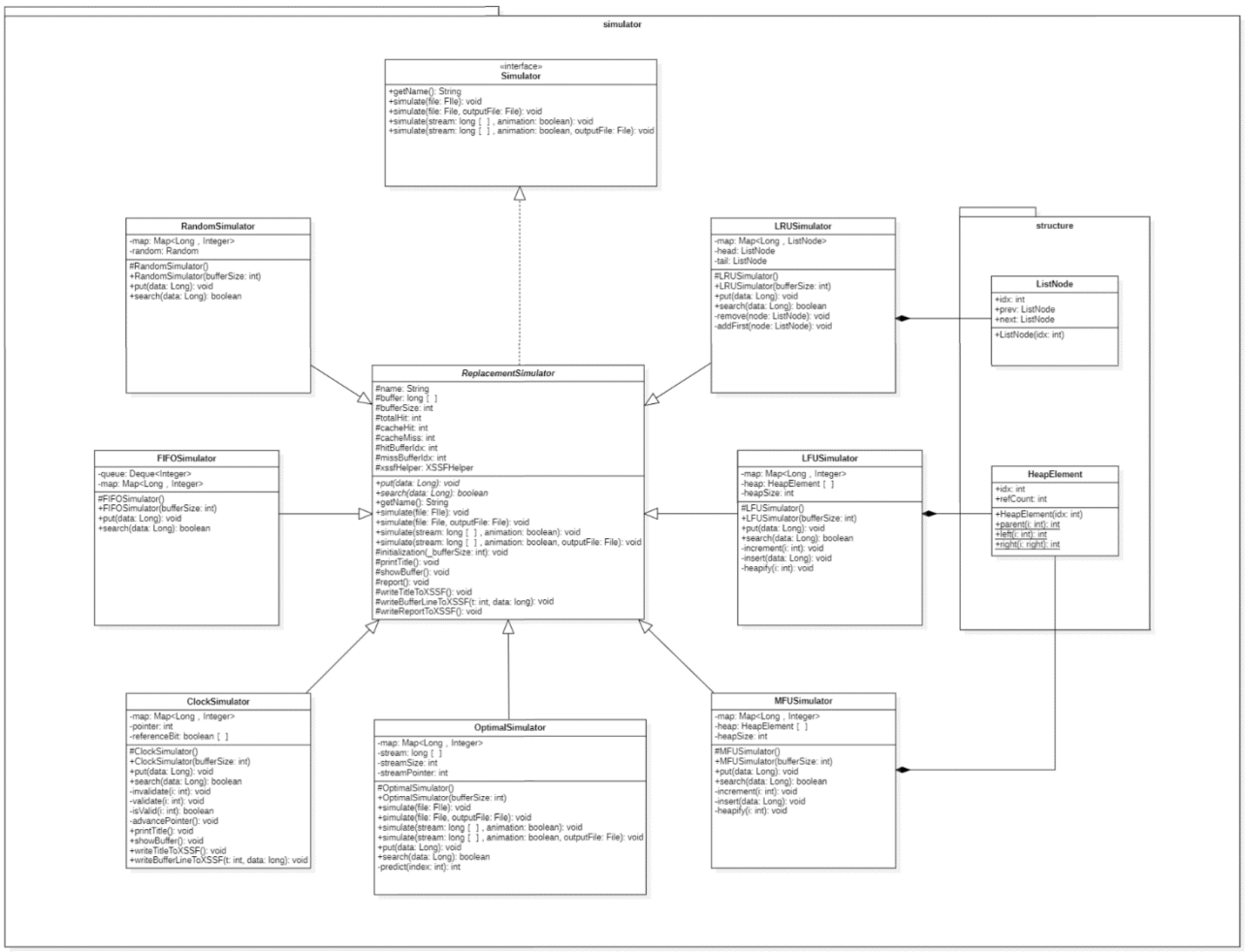
- poi-bin-5.2.3
  - https://www.apache.org/dyn/closer.lua/poi/release/bin/poi-bin-5.2.3-20220909.zip
- log4j-core-2.19.0
  - https://www.apache.org/dyn/closer.lua/logging/log4j/2.19.0/apache-log4j-2.19.0-bin.zip

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

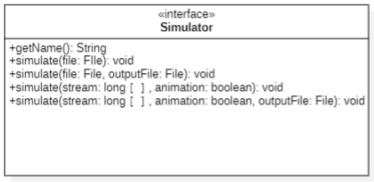


CLASS

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 Class 에서 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 • $O(n^2)$
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

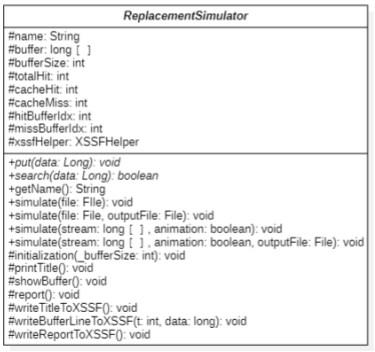
E: Enum I: 인터페이스 C: 클래스 (C): 추상클래스

SIMULATOR CLASSES DETAIL



Simulator 인터페이스

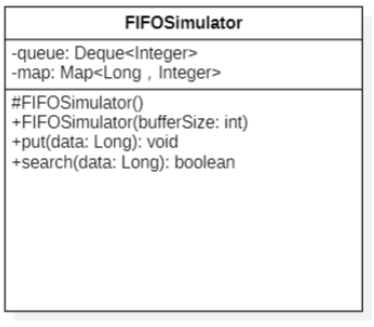
메소드	설명
getName	• 시뮬레이터 이름 반환
simulate	• 입력 데이터를 사용하여 페이지 교체 시뮬레이션 동작 • 4가지 오버로딩 - 파일로 데이터 입력 - 파일로 데이터 입력, 결과 파일 저장 - 배열로 데이터 입력 - 배열로 데이터 입력, 결과 파일 저장



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

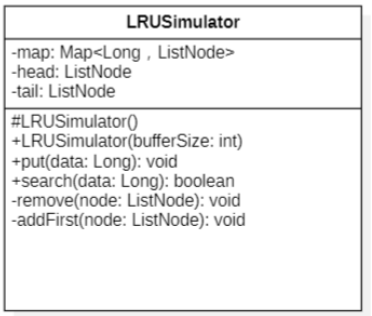
메소드	설명
put	• 데이터를 교체 정책에 따라 buffer에 입력 • 교체 정책마다 다르게 구현
search	• 입력된 데이터가 buffer에 있는지 판단
getName	• Simulator 인터페이스 구현
simulate	• Simulator 인터페이스 구현
initialization	• 필드 값 초기화
printTitle	• 애니메이션 결과 출력 시 상단 부분 문구 출력
showBuffer	• buffer 상태 출력
report	• 시뮬레이션의 결과 출력
writeTitleToXSSF	• XSSF 파일 셀에 상단 부분 입력
writeBufferLineToXSSF	• XSSF 파일 셀에 버퍼 내용 입력
writeReportToXSSF	• XSSF 파일 셀에 시뮬레이션 결과 입력



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

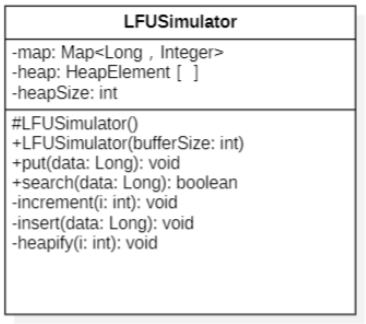


LRU Simulator

- 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

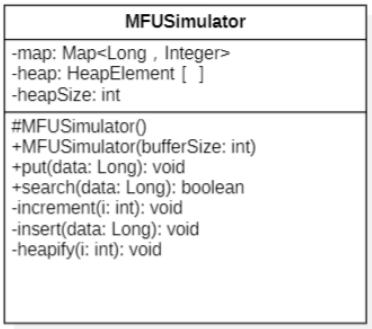


LFU Simulator

- 최소 힙으로 구현

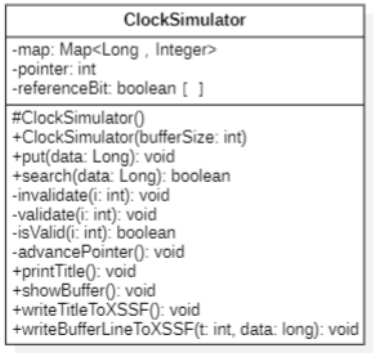
필드	설명
map	• buffer에 존재하는지 여부를 빠르게 확인하기 위한 맵
heap	• 힙 자료구조
heapSize	• 힙 크기

메소드	설명
increment	• 해당 인덱스의 원소 값을 1 증가시키고, 다시 힙 구성
insert	• 힙 삽입 연산
heapify	• heap 배열을 최소 힙 구조에 맞게 재배치



MFU Simulator

- 최대 힙으로 구현
- LFU Simulator와 동일

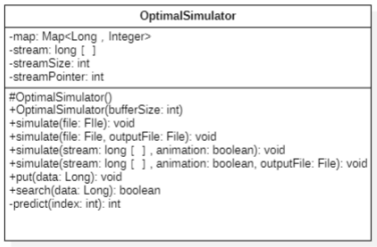


Clock Simulator

- 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

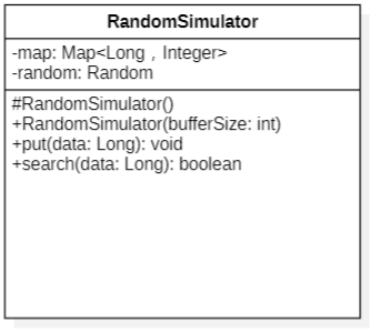


Optimal Simulator

- 매번 입력될 때 마다 predict 메소드로 교체할 페이지 선택
- predict 메소드가 $O(n^2)$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



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

PRODUCT

입력 예시

```
select data input mode. (0: direct input, 1: file upload, 2: use prev data, 99: terminate)
~ 0

enter positive number stream.
 * the number can not exceed 99999999
 * separated by blank
 * show animation
~ 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

enter buffer size.
~ 3

select page replacement policy.(multiple select possible)
 0: FIFO
 1: LRU
 2: LFU
 3: MFU
 4: Random
 5: Clock
 6: Optimal [O(n^2) : cannot operate if stream size > 10000]
~ 0 1 2 3 4 5 6

do you want animation? (1: yes, other: no)
~ 0

do you want output excel file? (1: yes, other: no)
~ 0
```

결과 출력

```
FIFO simulator report
  Buffer Size : 3
  Cache Hit   : 5
  Cache Miss  : 15
  Total Hit   : 20
  Hit Ratio   : 25.000000%

random simulator report
  Buffer Size : 3
  Cache Hit   : 7
  Cache Miss  : 13
  Total Hit   : 20
  Hit Ratio   : 35.000000%

LRU simulator report
  Buffer Size : 3
  Cache Hit   : 8
  Cache Miss  : 12
  Total Hit   : 20
  Hit Ratio   : 40.000000%

clock simulator report
  Buffer Size : 3
  Cache Hit   : 9
  Cache Miss  : 11
  Total Hit   : 20
  Hit Ratio   : 45.000000%

LFU simulator report
  Buffer Size : 3
  Cache Hit   : 7
  Cache Miss  : 13
  Total Hit   : 20
  Hit Ratio   : 35.000000%

optimal simulator report
  Buffer Size : 3
  Cache Hit   : 11
  Cache Miss  : 9
  Total Hit   : 20
  Hit Ratio   : 55.000000%

MFU simulator report
  Buffer Size : 3
  Cache Hit   : 8
  Cache Miss  : 12
  Total Hit   : 20
  Hit Ratio   : 40.000000%
```

애니메이션 출력(예시: optimal simulator)

optimal animation. (blue : hit, red : page fault)

time	stream	page1	page2	page3
1	7	7	X	X
2	0	7	0	X
3	1	7	0	1
4	2	2	0	1
5	0	2	0	1
6	3	2	0	3
7	0	2	0	3
8	4	2	4	3
9	2	2	4	3
10	3	2	4	3
11	0	2	0	3
12	3	2	0	3
13	2	2	0	3
14	1	2	0	1
15	2	2	0	1
16	0	2	0	1
17	1	2	0	1
18	7	7	0	1
19	0	7	0	1
20	1	7	0	1

엑셀 저장(예시: clock simulator)

	A	B	C	D	E	F	G
1							
2		clock simulator report					
3							
4		buffer size	3				
5		cache hit	9				
6		cache miss	11				
7		total hit	20				
8		hit ratio	45		hit	miss	reference
9							
10		time	stream	cursor	page 1	page 2	page 3
11		1	7	-1	7		
12		2	0	-1	7	0	
13		3	1	-1	7	0	1
14		4	2	0	2	0	1
15		5	0	0	2	0	1
16		6	3	2	2	0	3
17		7	0	2	2	0	3
18		8	4	0	4	0	3
19		9	2	2	4	0	2
20		10	3	0	3	0	2
21		11	0	0	3	0	2
22		12	3	0	3	0	2
23		13	2	0	3	0	2
24		14	1	1	3	1	2
25		15	2	1	3	1	2
26		16	0	0	0	1	2
27		17	1	0	0	1	2
28		18	7	2	0	1	7
29		19	0	2	0	1	7
30		20	1	2	0	1	7
31							

CLOCK_20221118
_0331585