“Page Replacement Simulator”

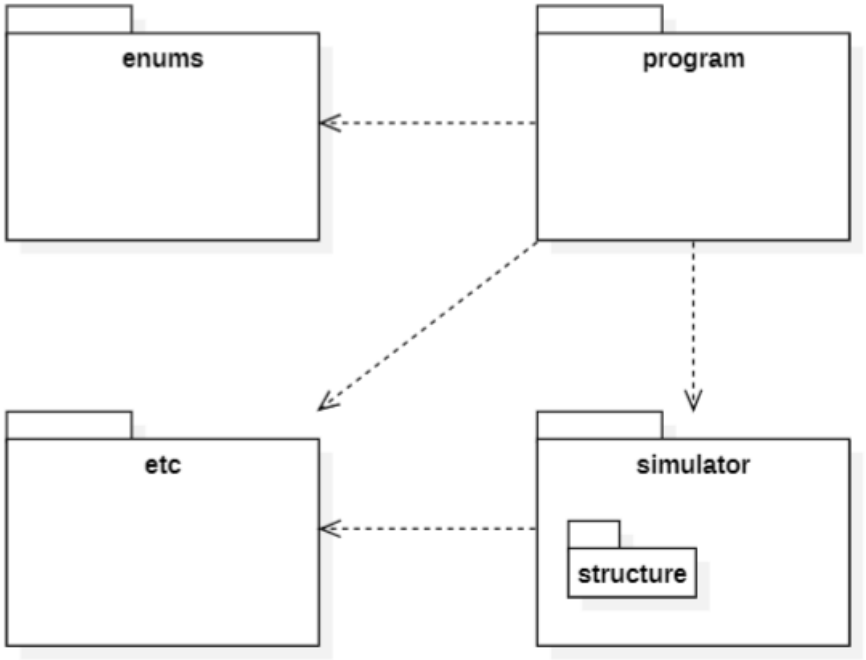
REQUIREMENTS

- 다양한 페이지 교체 정책 시뮬레이션 제공
- 데이터 스트림을 직접 입력하거나 파일로 입력
- 페이지 교체 애니메이션 제공
- Hit, Miss, Hit Ratio 결과 보고서 제공

DEVELOPMENT

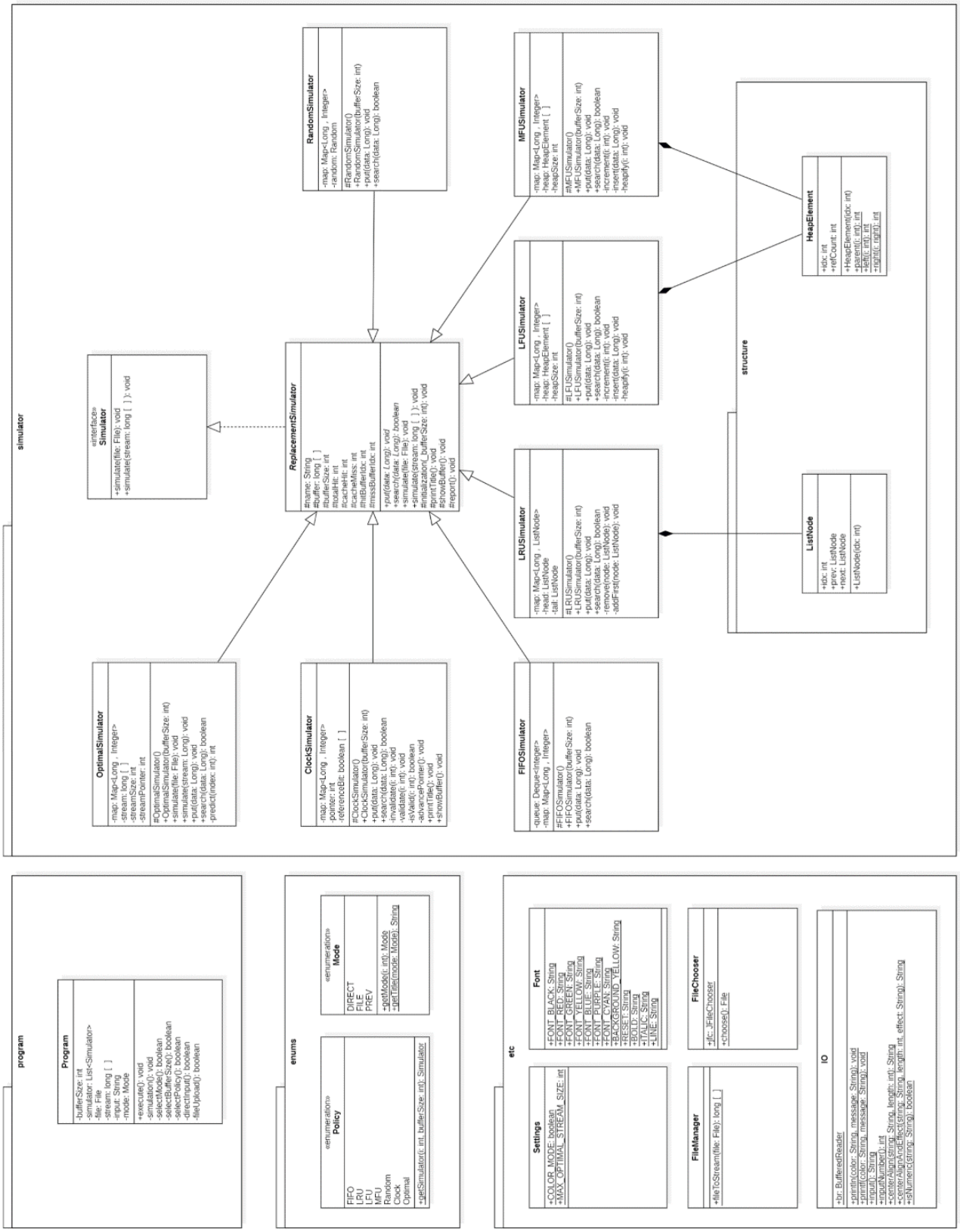
- Java
- IntelliJ

PACKAGES

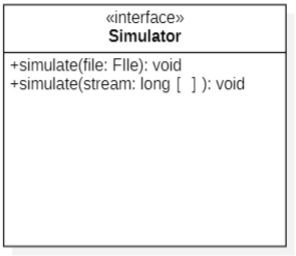


패키지	설명
program	Simulator를 동작시키는 Program Class을 제공하는 패키지
simulator	Simulator를 제공하는 패키지
structure	Simulator 구현에 필요한 자료구조 Class를 제공하는 패키지
etc	File, IO에 관련된 Class를 제공하는 패키지
enums	프로젝트에 관련된 Enum Class를 제공하는 패키지

CLASS

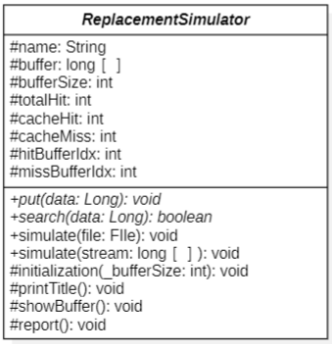

program
Program
-bufferSize: int
-simulator: List<Simulator>
-file: File
-stream: long [ ]
-input: String
-mode: Mode
+execute(): void
-simulation(): void
-selectMode(): boolean
-selectBufferSize(): boolean
-selectPolicy(): boolean
-directInput(): boolean
-fileUpload(): boolean
enums
«enumeration»
Policy
FIFO
LFU
LRU
MFU
Random
Clock
Optimal
+getSimulator(i: int, bufferSize: int): Simulator
«enumeration»
Mode
DIRECT
FILE
PREV
-getMode(i: int): Mode
-getTitle(mode: Mode): String
etc
Settings
+COLOR_MODE: boolean
+MAX_OPTIMAL_STREAM_SIZE: int
Font
+FONT_BLACK: String
+FONT_RED: String
+FONT_GREEN: String
+FONT_YELLOW: String
+FONT_BLUE: String
+FONT_PURPLE: String
+FONT_CYAN: String
+BACKGROUND_YELLOW: String
+RESET: String
+BOLD: String
+ITALIC: String
+LINE: String
FileChooser
-jfc: JFileChooser
+choose(): File
FileManager
+fileToStream(file: File): long [ ]
IO
-br: BufferedReader
+println(color: String, message: String): void
+printf(color: String, message: String): void
+input(): String
+inputNumber(): int
+centerAlign(string: String, length: int): String
+centerAlignAndSet(string: String, length: int, effect: String): String
+isNumeric(string: String): boolean
simulator
«interface»
Simulator
+simulate(file: File): void
+simulate(stream: long [ ]): void
ReplacementSimulator
#name: String
#buffer: long [ ]
#bufferSize: int
#totalHit: int
#cacheHit: int
#cacheMiss: int
#hitBufferIdx: int
#missBufferIdx: int
+put(data: Long): void
+search(data: Long): boolean
+simulate(stream: long [ ]): void
+simulate(file: File): void
+initialization(_bufferSize: int): void
#showBuffer(): void
#report(): void
RandomSimulator
-map: Map<Long , Integer>
-random: Random
#RandomSimulator()
+RandomSimulator(bufferSize: int)
+put(data: Long): void
+search(data: Long): boolean
OptimalSimulator
-map: Map<Long , Integer>
-stream: long [ ]
-streamSize: int
-streamPointer: int
#OptimalSimulator()
+OptimalSimulator(bufferSize: int)
+simulate(file: File): void
+simulate(stream: Long): void
+put(data: Long): void
+search(data: Long): boolean
-predict(index: int): int
ClockSimulator
-map: Map<Long , Integer>
-pointer: int
-referenceBit: boolean [ ]
#ClockSimulator()
+ClockSimulator(bufferSize: int)
+put(data: Long): void
+search(data: Long): boolean
-invalidate(i: int): void
-validate(i: int): void
-isValid(i: int): boolean
-advancePointer(): void
+printTitle(): void
+showBuffer(): void
FIFOSimulator
-queue: Deque<Integer>
-map: Map<Long , Integer>
#FIFOSimulator()
+FIFOSimulator(bufferSize: int)
+put(data: Long): void
+search(data: Long): boolean
LRUSimulator
-map: Map<Long , ListNode>
-head: ListNode
-tail: ListNode
#LRUSimulator()
+LRUSimulator(bufferSize: int)
+put(data: Long): void
+search(data: Long): boolean
-remove(node: ListNode): void
-addFirst(node: ListNode): void
LFUSimulator
-map: Map<Long , Integer>
-heap: HeapElement [ ]
-heapSize: int
#LFUSimulator()
+LFUSimulator(bufferSize: int)
+put(data: Long): void
+search(data: Long): boolean
-increment(i: int): void
-insert(data: Long): void
-heapify(i: int): void
MFUSimulator
-map: Map<Long , Integer>
-heap: HeapElement [ ]
-heapSize: int
#MFUSimulator()
+MFUSimulator(bufferSize: int)
+put(data: Long): void
+search(data: Long): boolean
-increment(i: int): void
-insert(data: Long): void
-heapify(i: int): void
structure
HeapElement
+idx: int
+refCount: int
+HeapElement(idx: int)
-parent(i: int): int
-left(i: int): int
-right(i: right): int
ListNode
+idx: int
+prev: ListNode
+next: ListNode
+ListNode(idx: int)

SIMULATOR CLASSES



시뮬레이션을 할 수 있는 simulate 메소드를 제공하는 인터페이스

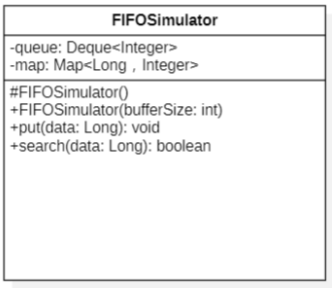
메소드	설명
simulate	• 입력 데이터를 사용하여 페이지 교체 시뮬레이션 동작 • File, long[] 형식으로 데이터를 입력할 수 있음



Simulator 구현체로 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	데이터가 buffer에 이미 존재하는 경우를 세는 카운터
cacheMiss	데이터가 buffer에 존재하지 않아 교체가 일어나는 경우를 세는 카운터
hitBufferIdx	hit가 일어난 buffer의 인덱스 기록(showBuffer 메소드에서 사용)
missBufferIdx	miss가 일어난 buffer의 인덱스 기록(showBuffer 메소드에서 사용)

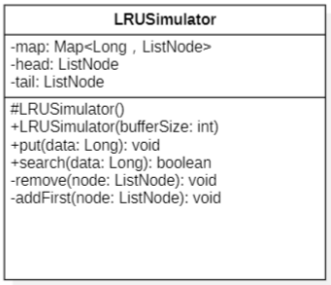
메소드	설명
put	• 데이터를 교체 정책에 따라 buffer에 입력 • 교체 정책마다 다르게 구현
search	입력된 데이터가 buffer에 있는지 판단
initialization	필드 값 초기화
printTitle	애니메이션 결과 출력 시 상단 부분 문구 출력
showBuffer	buffer 상태 출력
report	시뮬레이션의 결과 출력



FIFO Simulator 구현

- 입력된 순서를 저장하고 있는 큐 자료구조로 구현

필드	설명
queue	입력된 순서를 유지하는 큐 자료구조
map	데이터가 buffer에 존재하는지 여부를 빠르게 확인하기 위한 맵

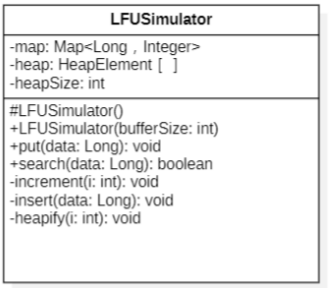


LRU Simulator 구현

- 이중 연결리스트로 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	데이터가 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

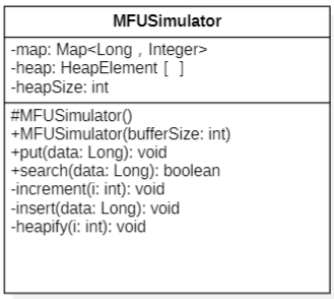


LFU Simulator 구현

- 최소 힙으로 구현

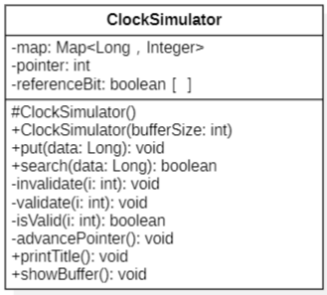
필드	설명
map	데이터가 buffer에 존재하는지 여부를 빠르게 확인하기 위한 맵
heap	힙 자료구조
heapSize	힙 크기

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



MFU Simulator 구현

- 최대 힙으로 구현
- 나머지는 LFU Simulator와 동일

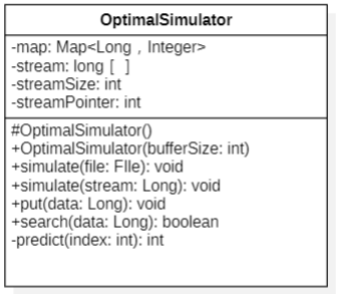


Clock Simulator 구현

- 원형 큐 방식으로 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	데이터가 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 출력 양식이 다르기 때문에 오버라이딩
showBuffer	다른 시뮬레이터와 출력 양식이 다르기 때문에 오버라이딩

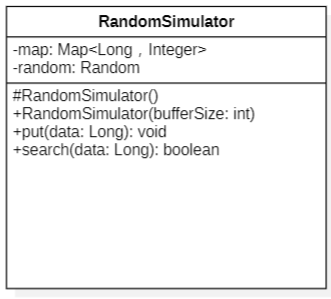


Optimal Simulator

- 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	데이터가 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
predict	앞으로 가장 오랫동안 사용되지 않을 buffer의 인덱스를 반환



Random Simulator 구현

- 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

- PageReplacementSimulator.jar