|  |
| --- |
| **[511643] 자료구조** |
| **실습 #10 보고서** |

|  |  |
| --- | --- |
| **이름** | 조아현 |
| **학번** | 20215247 |
| **소속**  **학과/대학** | 소프트웨어융합대학  빅데이터전공 |
| **분반** | 01 (담당교수: 김태운) |

## <주의사항>

* 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
* 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.
  + 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.
* SmartLEAD 제출 데드라인:
  + 2주 뒤 실습시간 전날(다음 다음번 실습 전날) 23:55까지
  + 데드라인을 지나서 제출하면 0점(예외 없음)
  + 주말/휴일/학교행사 등으로 인한 데드라인 연장 없음
  + 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0점 처리함
* SmartLEAD에 아래의 파일을 제출 해 주세요
  + 보고서(PDF 파일로 변환 후 제출 권장하나, WORD 형식으로 제출도 가능)
  + 보고서 파일명에 이름과 학번을 입력 해 주세요.
  + 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일을 하나의 zip 파일로 압축하여 제출(또는 본 문서에 소스코드 화면 캡쳐해서 붙여넣기)

## <개요>

이번 과제는 이진 탐색 트리 (Binary Search Tree, BST)를 구현하고 활용하는 내용입니다.

## <실습 과제>

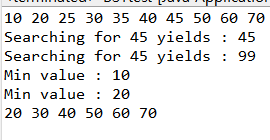
|  |
| --- |
| **[Q 0] 요약 [10 점]**  이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3문장으로 요약하세요. |

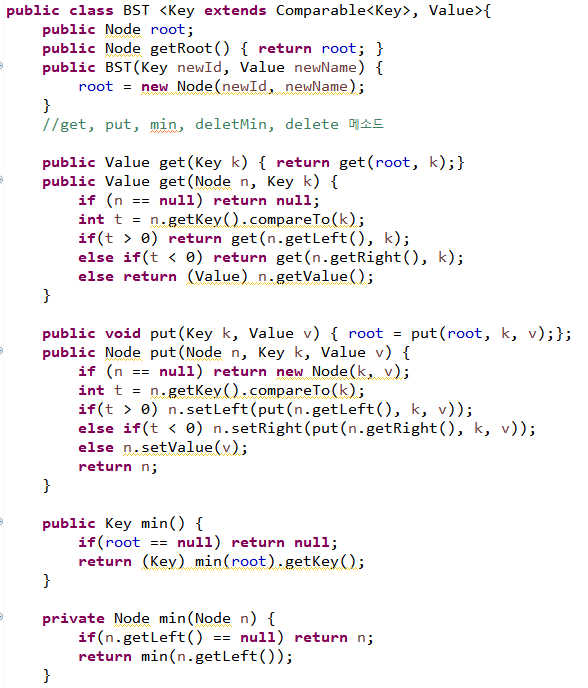
답변:

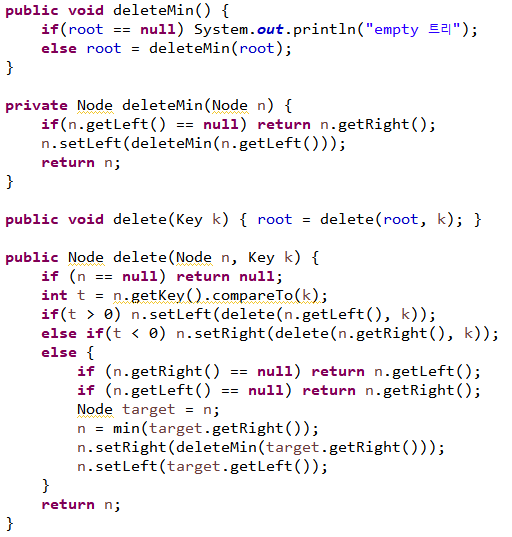
이진 탐색 트리는 정렬된 데이터의 중간에 위치한 항목을 기준으로 데이터를 두 부분으로 나누어 가며 특정 항목을 찾는 탐색 방법인 이진 탐색 개념을 트리형태의 구조에 접목한 자료 구조입니다. 이진 탐색 트리는 각 노드 n의 키 값이 n의 왼쪽 서브 트리에 있는 노드의 키 값보다 크고, n의 오른쪽 서브 트리에 있는 노드의 키 값보다는 작은 조건을 만족해야 합니다. 따라서 진 탐색 트리는 중위 순회하면 정렬되어 출력되는 특징을 가지고 있습니다.

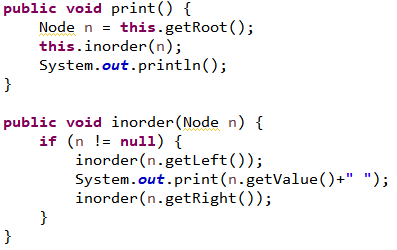
|  |
| --- |
| **[Q 1] BST 구현 및 테스트 코드 구현 [60점]**  강의자료와 동일하게 Node 클래스와 BST 클래스를 구현하세요. BST 클래스를 테스트하기 위한 BSTtest 클래스를 만들고, main 메소드를 강의노트 pp.37 ~ 38과 같이 구현하세요. [1]번~[8]번 과정을 모두 수행한 후, 터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다. |

답변:

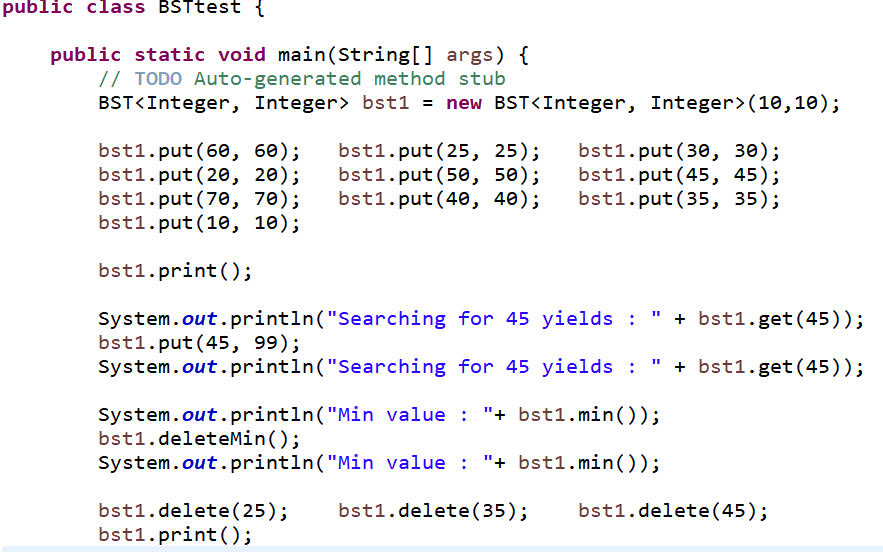


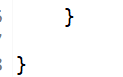






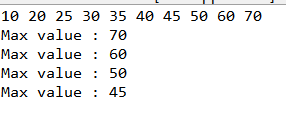


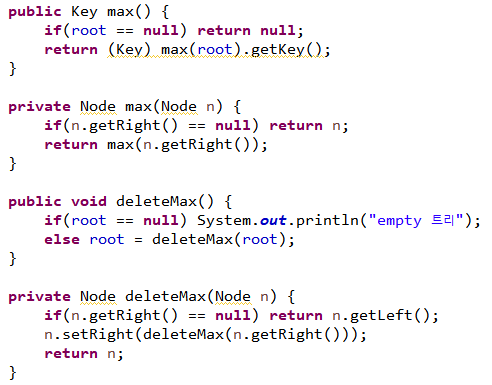


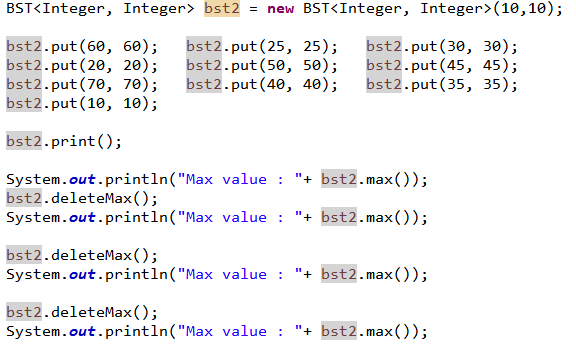


|  |
| --- |
| **[Q 2] delete max, 최대의 키값을 가지는 노드 삭제하기 [10점]**  BST 클래스에서 최댓값(Key 값 기준)을 삭제하는 public void deleteMax() 메소드를 구현하세요 (BST 클래스의 멤버 메소드로 구현). 강의자료 p.37을 참고하여 bst1 트리의 초기구성과 동일하게 bst2를 생성하세요. 초기의 트리 구성을 확인하기 위해 bst2.print 메소드를 호출하세요. bst2 트리에서 최댓값을 삭제하고 print 메소드를 호출하는 과정을 3회 반복하세요. 터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.  \*\* 힌트: find max => delete? |

답변:

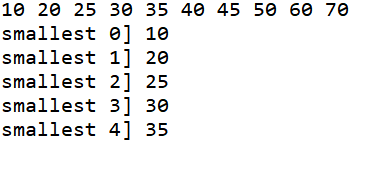


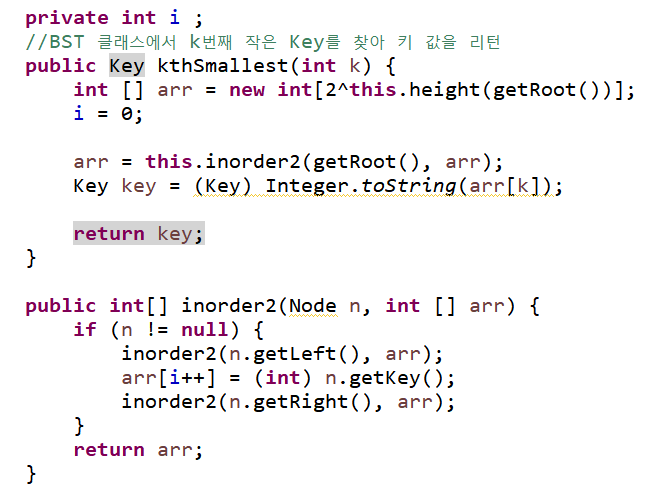


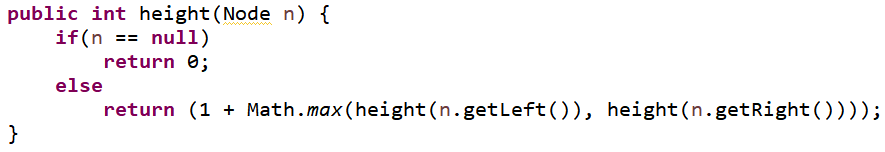


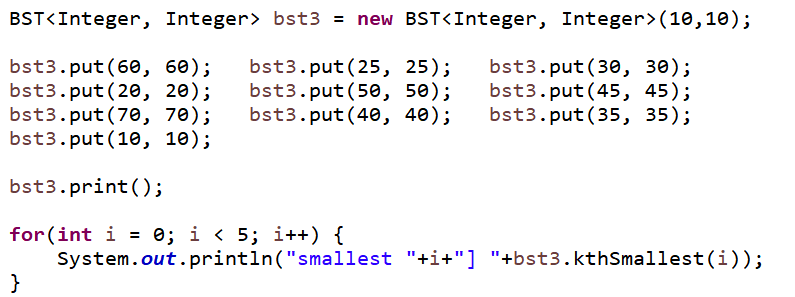
|  |
| --- |
| **[Q 3] k-th smallest node의 key 값 리턴하기 [10점]**  BST 클래스에서 k번째 작은 Key를 찾아 키 값을 리턴하는 public Key kthSmallest(int k) 메소드를 구현하세요 (BST 클래스의 멤버 메소드로 구현). 단, 가장 작은 키는 0번째이고, 두 번째로 작은 키가 1번째 입니다.  강의자료 p.37을 참고하여 bst1 트리의 초기구성과 동일하게 bst3를 생성하세요. bst3 트리에서 아래의 코드를 실행하세요  for(i=0;i<5;i++) 터미널에 bst3.kthSmallest(i)의 리턴 값 출력;  터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.  \*\* 힌트: 중위 순회 결과...? |

답변:



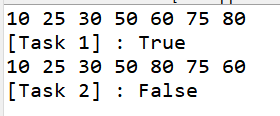


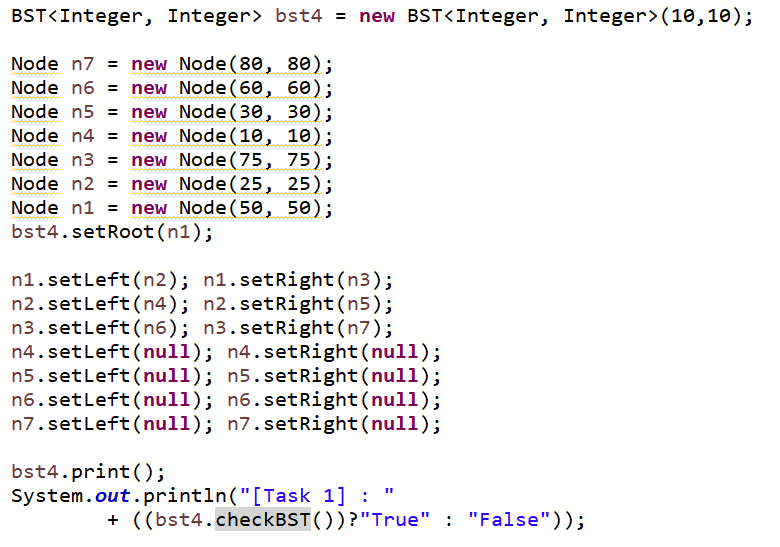




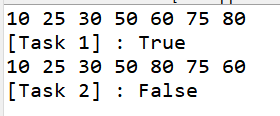
|  |
| --- |
| **[Q 4] check BST [10점]**  주어진 BST 클래스 인스턴스가 이진 탐색 트리가 맞는지를 검사하는 public boolean checkBST() 메소드를 BST 클래스의 멤버 메소드로 구현하세요. 아래와 같은 순서로 문제를 해결하세요.   * 아무런 인자를 받지않는 BST 클래스의 생성자 추가 * BST 클래스에 setRoot 메소드 추가 * BST.put 메소드를 사용하지 않고, 클래스 외부(즉, main 메소드)에서 Node를 이어서 트리를 구성한 후, 트리의 root를 인자로 BST 클래스의 setRoot 호출   + (노드를 직접 연결하는 방식으로) 이진 탐색 트리 속성을 만족하는 트리(bst4) 만들기   + (노드를 직접 연결하는 방식으로) 이진 트리 이지만 이진 탐색 트리의 속성을 만족하지 않는 트리(bst5) 만들기   + 문제에서 사용할 두 개의 트리(bst4, bst5)는 강의노트 p.40를 참고.   [Task 1] bst4 이진 탐색 트리를 생성하고, print() 메소드를 호출하세요. bst4.checkBST()를 호출하고, 결과를 터미널에 출력하세요. 터미널 화면을 캡처하고 본 문서에 첨부하세요.  [Task 2] bst5 이진 탐색 트리를 생성하고, print() 메소드를 호출하세요. bst5.checkBST()를 호출하고 결과를 터미널에 출력하세요. 터미널 화면을 캡처하고 본 문서에 첨부하세요.  소스코드도 제출해야 합니다.  \*\* 힌트: 중위 순회 결과...? |

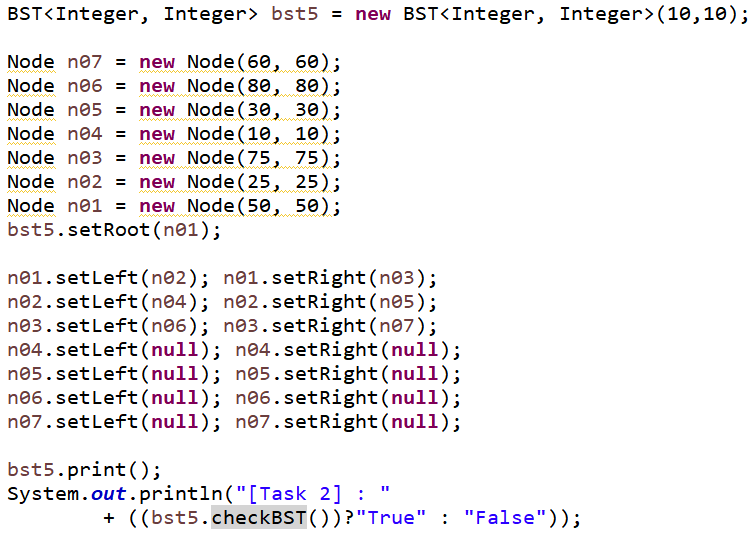
답변 [Task 1]:

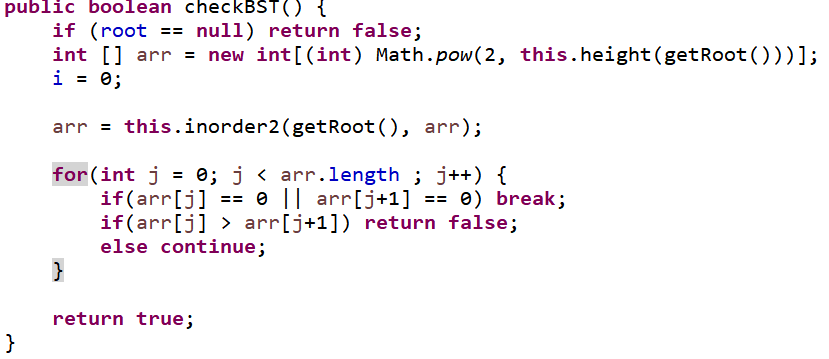


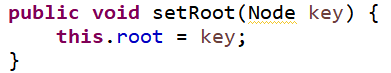


답변 [Task 2]:









**끝! 수고하셨습니다 ☺**