

Chapter 05. 손코딩 관련 주요 문항 및 단계별 답변 팁

손코딩 관련 주요 문항 및 단계별 답변 팁

목표: 신입 기술면접 준비 방법 익히기

- 기술 면접의 유형과, 문항 유형에 대해 정리하고,
- 각 문항 유형별 예시와 답변 예시를 알아보고,
- 답변 가이드 라인과 팁을 정리하며,
- 각 기술면접 유형과, 문항 유형 준비 방안에 대해 소개

문제 유형

1. 문제 유형1 - 자기 소개 관련
2. 문제 유형2 - CS 기본 지식 관련
3. 문제 유형3 - 프로그래밍 언어별/각 IT 분야별 세부 지식 관련
4. 문제 유형4 - 자료구조/알고리즘 기반 문제 손코딩

각 유형별 주요 문제/예시 답변/팁 알아보기

- 왜 그런 문제를 물어보나?
- 어떤 답변을 기대하는가?
 - 예시 질문 및 답변
- 팁은 무엇인가?

손코딩 테스트 이유

- 기술적 커뮤니케이션 역량
- 효율적인 프로그래밍 작성 역량
- 꼼꼼한 프로그래밍 작성 역량

단순히 문제를 프로그래밍으로 풀었는지를 확인하는 것이 아님

주요 손코딩 테스트 검증 항목 (참고)

1. hands-on coding ability
2. logical thinking
3. code style
4. code quality
5. corner case sense
6. ability to provide different solution
7. communication ability
8. data structure/algorithm knowledge
9. complexity estimation ability

손코딩 테스트시 단계

1. 문제 듣기
2. 예제를 통해 문제 이해하기
3. 무식하게 풀기
4. 질문자와 커뮤니케이션을 통해 개선하기
5. 마이너한 부분의 최적화
6. 검토하기

손코딩 테스트시 단계별 유의점

1. 문제 듣기

- 문제 설명 관련 정보는 하나라도 놓치지 않고 들어야 함
 - 세세한 문구 하나하나에 조건이 숨겨져 있을 수 있음
 - 조건을 놓치면 요구되지 않은 프로그램을 작성할 수 있음
 - 애매하게 이야기한 부분은 반드시 짚고 넘어가야 함
 - 의도적으로 애매하게 이야기해서, 조건을 명확하게 커뮤니케이션을 통해 정의할 수 있는지를 판단함

손코딩 테스트시 단계별 유의점

2. 예제를 통해 문제 이해하기

- 직접 예제를 적어보고, 문제를 이해하고, 숨은 조건을 명확히 커뮤니케이션을 통해 확인함

3. 무식하게 풀기

- 처음부터 최적의 알고리즘을 생각하지 말 것
- 일단 무식한 코드라도 좋으니, 코드로 작성해서 요구에 맞는 결과를 낼 것
- 시간 복잡도를 대략적으로 계산해서 알려줄 것

손코딩 테스트시 단계별 유의점

4. 질문자와 커뮤니케이션을 통해 개선하기

- 개선의 포인트를 질문자와의 커뮤니케이션을 통해 찾아내기
- 개선할만한 부분의 코드를 수정하기
- 개선된 시간 복잡도를 대략적으로 계산해서 알려줄 것

손코딩 테스트시 단계별 유의점

5. 마이너한 부분의 최적화

- 병목현상, 불필요 작업, 중복 작업, 놓친 부분을 최적화하며 완벽한 코드를 만들 것
- 방어코드를 넣을 것

6. 검토하기

- 몇가지 예제를 가지고 간단히 테스트해보고 문제 없음을 보여줄 것

무엇보다 본 강좌가 꼭 도움이 되셨으면 좋겠습니다.

또한 본 강좌를 수강하신 후, 긍정적인 피드백이 있으시다면,
가감없이 언제든지 해주신다면, 저에게 정말 큰 힘이 되오니, 긍정적인 피드백 부탁드립니다.

Dave Lee 강사, 운영 블로그: 잔재미코딩(www.fun-coding.org)