



### 3. 코드 검사를 정의하고 그 목적을 정리하세요.

• 검사 = 품질 문제를 개인이 의지하지 않고 조직적으로 해결하려는 노력

• 코드 검사 = 높은 품질의 소프트웨어를 얻기 위한 품질보증 활동 중 하나로, 프로그램 만든 후 이미 대한 검사를 수행하는 것. → 코드 검사는 보다 자유로워야 함.

• 코드 검사 목적.

- ① 소프트웨어 개발과정에서 요구되는 공식 기술 검토 중 하나.
- ② 방대한 검사팀(4~5명)이 만들어져 요구사항과 표준에 따라 만들어졌는지 조사
- ③ 오류를 걸러내고, 다른 사람과 지식을 공유, 좋은 품질 시스템 개발의 공감대 형성을 하게 해줌.
- ④ 단위시험 이전, 이후에 이루어짐.
- ⑤ 프로그램의 스타일을 표준화 시킴.
- ⑥ 코드를 개발과정의 문서로 인식될 수 있게 함.
- ⑦ 품질보증 활동의 일부로 인식하게 함.
- ⑧ 프로그램 개발이 개인이 아닌 조직적으로 이루어짐을 보여줌.
- ⑨ 코드 검사자들은 각자 역할이 주어지며 사전에 코드에 대해 미리 조사해야 함 → 결과는 즉후 본리를 위해 문서기록
- ⑩ 동료 프로그래머들에 의해 이루어지는 것이 일반적.
- ⑪ 신중하게 코드에 대한 기술적 검토만 해야 함.
- ⑫ 검사를 위한 체계적인 틀이 만들어져야 함, 결과는 통계적인 자료로 수집되어 관리자에게 보고.

### 4. 코드의 오류 유형을 쓰고, 각 기능에 대해서 설명하세요.

• 코드 오류 유형

- ① 데이터 오류 : 데이터 다루는 부분에서의 오류로, 데이터유형, 정의, 변수선언, 대개변수에서 나타나는 오류
- ② 문서 오류 : 프로그램 구성 요소인 선언 부분, 서브루틴, 모듈에 대해 정확하지 않은 주석에 대한 오류
- ③ 기능 오류 : 서브루틴이나 블록이 잘못된 것(what)에 대한 오류.
- ④ 논리 오류 : 서브루틴이나 블록이 수행하는 방법(How)에 대한 오류.
- ⑤ 성능 오류 : 프로그램을 수행하며 요구되는 효율성(성능)을 만족시키지 못한 경우 발생하는 오류.
- ⑥ 표준 오류 : 구문이나 표현이 표준(일반적)과 다른 경우 발생하는 오류.
- ⑦ 기타 오류 : 문법적 오류, 사람의 개성이 포함된 오류, 등 앞의 유형으로 설명하기 힘든 불투명하고 애매모호한 것에 대한 오류.

• 오류의 종류

- ① 실증 : 구성 요소 안에 있어야 할 것이 없는 경우
- ② 실수 : 구성 요소 안에 있으나 잘못된 발견된 경우
- ③ 불필요 : 요구되는 것 이상으로 필요 없는 경우.