|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | 2019.10.01 | 학번 | 201644008 | 성명 | 박규민 |

|  |
| --- |
| 1. 교재 Chap 04 의 예제를 구현해보고 결과의 의미를 한줄이상 서술하시오  4-4, 6, 7, 9, 11~15, 16 |
| 결과에 한줄이상 설명 하기 |
| 4-4 url의 프로토콜, 증명, 호스트, 포트 등등 url의 정보가 표현된다.  4-6 null prototype { isbn : ‘~’}이 나오는데 isbn에 맞게 리턴받은 것 같다.  4-7 4의 예제랑 비슷하게 파싱되서 출력된다.  4-9 println과 비슷한 기능인 것 같다. // 52 + 273 = 325  4-11 text의 내용이 출력된다. // qwerty  4-12 11이랑 결과는 같지만 함수로 호출되므로 변화를 표현할수없다. // qwerty  4-13 text파일에 Hello world가 추가되었다.  qwerty  hello world  4-14 DB sql을 하는듯이 동기화로 예외처리하여 파싱을 하였다.  qwerty  hello world  4-15 try-catch가 아닌 함수이기 때문에 if로 예외처리를 하였다.  qwerty  hello world  4-16 다 false가 나왔다. 책하고 다르지만 true가 1이기 때문에 그렇거같다. // false |

|  |
| --- |
| 2. 교재 Chap 05 의 예제를 구현해보고 결과의 의미를 한줄이상 서술하시오  5-5, 7, 9, 10, 12,13, 16,17 |
| 실행코드, 실행결과, 결과의 의미에 대하여 한줄이상 설명 하기 |
| 5-5 에러를 발생시켜서 예외처리를 테스트하고 exit에 대한 이벤트도 구현하였다.  예외가 발생  예외가 발생  예외가 발생  예외가 발생  잘가요  5-7 이벤트의 개수를 늘려서 원래의 10개 제약을 풀었다. // 에러가 안뜬다  5-9 에러에 대한 이벤트를 1번 실행시키고 이벤트를 제거하고 다시 구현해서 프로그램 종료를 시켰다.  예외가 발생  5-10 이벤트가 1번만 실행되게 구현되어서 1번 실행 후 에러가 발생하자 프로그램 종료가 되었다.  예외가 발생  5-12 이벤트가 강제로 실행되었지만 메소드가 아닌 이벤트만 발생시킨다.  예외가 발생  예외가 발생  예외가 발생  예외가 발생  실행중  예외가 발생  5-13 강제 실행과 실제 실행의 차이에 대해 알 수 있습니다.  잘가요  5-16, 17 이벤트 객체를 만들고 실행하여 테스트 해봤습니다.  이벤트 실행  이벤트 실행  이벤트 실행  이벤트 실행  이벤트 실행 |

|  |
| --- |
| 2. 교재 Chap 04 의 예제를 구현해보고 결과의 의미를 한줄이상 서술하시오  코드4-2 |
| 실행코드, 실행결과, 결과의 의미에 대하여 한줄이상 설명 하기 |
| Var os = require(‘os’)를 하게 되면 시스템 정보를 모듈로 불러오게된다.  Hostname은 현재 pc의 호스트명을 리턴합니다. //Rigel  Type은 운영체제 명을 리턴합니다. // windows.\_NT  Platform은 운영체제 플랫폼을 리턴합니다. // win32  Arch는 아키텍처를 리턴합니다. // x64  Release는 운영체제 버전을 식별하는 문자열을 리턴합니다. // 10.0.17763  Uptime은 실행된 시간을 리턴합니다. // 179910  Loadavg은 시스템 활동의 측정치 입니다. // [0, 0, 0]  Freemem은 사용안하는 유휴 메모리를 리턴합니다. // 3375480832  Cpus은 cpu 정보를 담은 객체를 리턴합니다. // model: 'Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz..  networkInterfaces은 네트워크 주소가 지정된 인터페이스를 리턴합니다.  // Wi-Fi': [{ address: 'fe80::348b:add3:9aa1:1b35', …… |