

11. API(Application Programming Interface)

//src: ch14_api 참고

1) String (20-12-11자료 참고)

(1) StringBuffer와 StringBuilder

- 문자열 변수의 변경이 많을 경우 String은 새로운 메모리를 생성하므로 속도가 느려짐
- 문자열 변경을 자주해야 할 경우 StringBuffer나 StringBuilder가 적합함
- 객체 내부에 있는 버퍼(buffer, 데이터를 임시로 저장하는 메모리)에 문자열의 내용을 저장해 두고 그 안에서 추가, 수정, 삭제 작업을 하므로 새로운 객체를 만들지 않고 문자열 조작이 가능
- 속도적인 측면: (빠르다) StringBuilder > StringBuffer > > > > String(느리다)

※ StringBuilder 주요 기능(method)

- `append(String str)`: 문자열 str 추가
- `insert(int index, String str)`: 특정 index에 문자열 str 추가
- `delete(int start, int end)`: index위치 start부터 end앞 까지 삭제
- `deleteCharAt(int index)`: index위치의 특정 문자 하나 삭제
- `int capacity()`: 문자열 크기 반환
- `ensureCapacity(int size)`: 버퍼의 크기를 size만큼 늘리는 메소드
- `trimToSize()`: 과도한 버퍼 크기를 적당하게 줄이는 메소드

※`System.currentTimeMillis()`; 1970년~현재 밀리세컨(1/1,000초) 단위로 표시, 속도 테스트시 사용※`StringTokenizer`; 문자열 분할 `//space, \t, \n`기준으로 token나눔, 그 밖의 기준은 입력 ex) "/"**2) 날짜(Calendar, GregorianCalendar, Date)** //src: ch14_api(com.lec.ex03_date) 참고(1) **Calendar**: 날짜와 시간을 표현할 때 많이 사용, SingletonClass (new 생성자 사용 불가)`Calendar calendar = Calendar.getInstance();``int year = calendar.get(Calendar.YEAR);``int month = calendar.get(Calendar.MONTH) + 1; //0~11월이라 현실이랑 맞춰주기 위해 +1`(2) **GregorianCalendar**: 날짜와 시간을 표현할 때 많이 사용, 일반 클래스`GregorianCalendar gc = new GregorianCalendar();``int month = gc.get(Calendar.MONTH) + 1; //0~11월이라 현실이랑 맞춰주기 위해 +1`(3) **Date**① `Date date = new Date();``int year = date.getYear() + 1900; int month = date.getMonth() + 1;`② `Date date = new Date(new GregorianCalendar(2020, 11, 14)); //2020년 12월 14일`