20-12-11 Java 스트레티지 패턴과 스트링(Pattern_Strategy pattern&API_String) ⓒcherryuki(ji)

10. 패턴 (Pattern) //src: ch13_pattern 참고

1) 싱글톤 패턴(Singleton pattern) (20-12-10자료 참고)

2) 스트레티지 패턴(Strategy pattern)

(1) Strategy pattern

; 각각의 기능을 부품화, 표준화해 캡슐처럼 교환해서 사용할 수 있게 만든 패턴

- 유지보수하기 용이하게 만들어진 유연한 프로그램(예: 신제품 출시, 기존 모델 업그레이드)

① 1단계: 개별 클래스 정의

② 2단계: 각 클래스의 공통점을 슈퍼클래스로 만들고 클래스 상속

③ 3단계: 공통점을 추상클래스로 추상화

④ 4단계: 각각의 기능을 객체(부품)화; **Object modularization**

(인터페이스를 활용하여 각각의 기능들을 부품화하고 묶어줌)

11. API(Application Programming Interface) //src: ch14_api 참고

1) String; 문자열에 관련된 대표적 클래스

- 객체 자료형(객체 데이터) but, 생성자(new) 사용하지 않고 초기화 가능
- 메모리를 과소비(단점); 초기화 된 데이터에 변화가 생기면 기존 것 버리고 새 메모리 이용

※String 주요 기능들(method)

- String concat(String str): 저장된 문자열과 str문자열을 결합
- String substring(int begin): 시작 위치부터 마지막까지의 문자열을 반환
- int length(): 문자열 길이를 반환
- String toUpperCase(): 대문자로 반환
- String toLowerCase(): 소문자로 반환
- char charAt(int index): index 위치의 문자를 반환
- int indexOf(char ch): 첫번째 ch문자의 위치를 반환
- int lastIndexOf(char ch): 마지막 ch문자의 위치를 반환
- boolean equals(String str): 지정된 문자열과 str문자열이 같은지 비교(대소문자 구분: A≠a)
- boolean equalsIgnoreCase(String str): 지정된 문자열과 str문자열이 같은지 비교(A=a)
- String trim(): 문자열 앞뒤 공백제거
- String replace(char old, char new): 문자열 내의 old문자를 new문자로 반환
- String repalceAll(String old, String new): 문자열 내의 old문자열을 new문자열로 반환