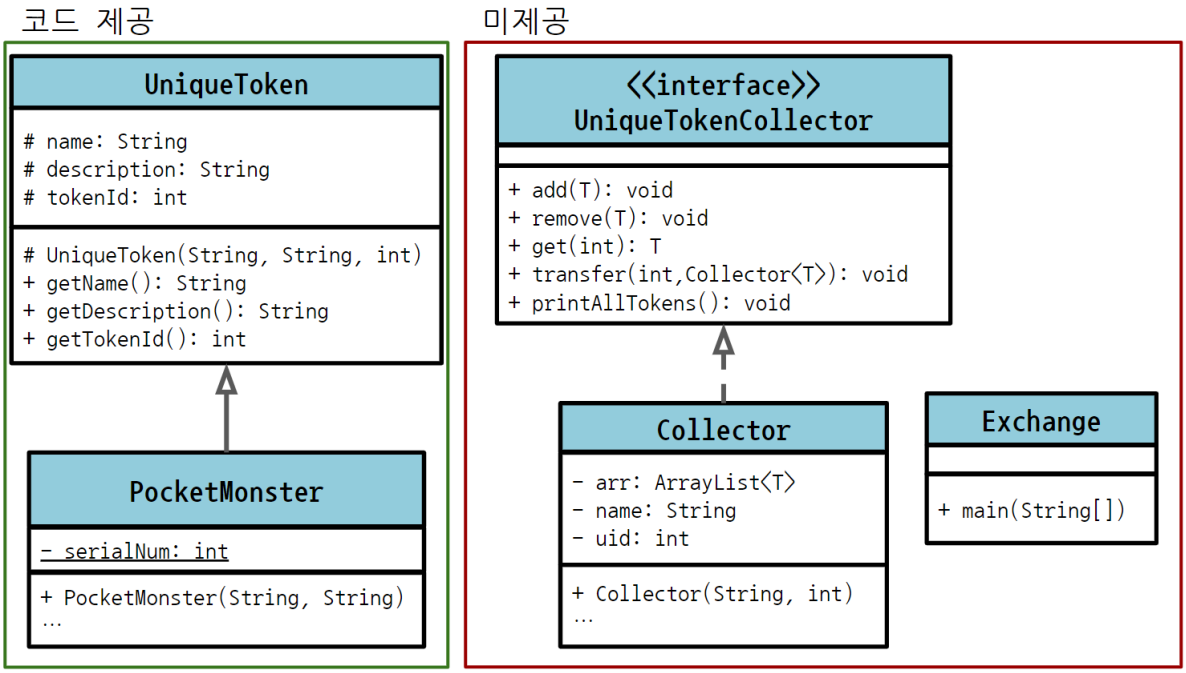
퀴즈 #2 (수목반)

특정 UniqueToken만 거래하는 Collector들 간 token 교환을 위한
적절한 인터페이스 및 제네릭 클래스를 생성한다.

퀴즈 제출시 LMS에 실행 결과 이미지만 업로드 하고
소스코드는 GitLab 으로 제출 바랍니다.



quiz2 패키지 아래에 다음과 같은 클래스들을 생성한다.

UniqueToken 클래스,
UniqueToken 을 상속받는 PocketMonster 클래스,
UniqueTokenCollector 인터페이스,
UniqueTokenCollector 구현하는 Collector 클래스,
Exchange 클래스

UniqueToken

- protected String name // 토큰의 이름
- protected int tokenId // 토큰 id
- protected String description // 토큰 정보

```
public class UniqueToken {
	protected String name;
	protected String description;
	protected int tokenId;

	protected UniqueToken(String n,String d,int t){
		name=n;
		description=d;
		tokenId=t;
	}
	public int getTokenId() {
		return this.tokenId;
	}
	public String getName() {
		return this.name;
	}
	public String getDescription() {
		return this.description;
	}

}
```

PocketMonster

static int serialNum = 0; // 객체 생성시 1씩 증가

PocketMonster(String name, String desc)

- name, description은 인자 받아서 초기화
- tokenId는 객체 생성시 serialNum 1씩 증가된 값 할당
- 첫 객체는 1부터 시작

```
public class PocketMonster extends UniqueToken {
	static private int serialNum=0;

	public PocketMonster(String name, String desc){
		super(name,desc,++serialNum);
	}
}
```

UniqueTokenCollector (interface)

- **UniqueToken 을 상속받는 클래스만 생성할 수 있는** 제네릭 타입 파라미터를 갖는다.
- void add(T item); // 인자로 들어온 토큰을 ArrayList에 추가
- void remove(T item); // 인자로 들어온 토큰을 ArrayList에서 제거

- T get(int id); // 현재 가진 토큰 중에 입력 받은 토큰 ID를 가진 토큰 반환, 없으면 null
- void transfer(int id, Collector<T> peer); // 아래와 같은 로직을 수행한다.
    1. 입력 받은 토큰 ID를 가진 토큰을 찾고
    2. 해당 토큰을 가지고 있을 시
        a. 토큰의 이름을 출력한 뒤
        b. 본인이 가진 토큰을 지우고
        c. 상대 객체에 해당 토큰을 추가
        d. 전송 완료 출력

- void printAllTokens(); // 아래와 같은 문구를 출력한다. (출력 예시 참고)
    ${Collector_Name}(${Collector_User_ID}):
    " 해당 유저가 가진 토큰 목록을 이름, ID, 설명 순서로 출력 "

Collector

- **UniqueToken 을 상속받는 클래스만 생성할 수 있는** 제네릭 타입 파라미터를 갖는다.
- private ArrayList<T> arr; // 본인이 가진 토큰의 목록
- private String name; // 본인 이름
- private int uid; // 본인 ID
- 인터페이스에 적힌 설명을 여기에서 구현한다.

Exchange

main 메소드 설명

- PocketMonster 타입 파라미터를 갖는 Collector 객체를 2개 생성한다.
각각 초기값은 다음과 같다.
name: " Red ", uid: 100
name: " Green ", uid: 200
- PocketMonster 객체를 2개 생성한다. 각각 초기값은 다음과 같다.
name: " Pikachu ", description: " 1 Million Volt Electric Shock "
name: " GGobugi ", description: " Surfing "

첫번째 Collector 객체에 첫번째 PocketMonster 객체를 추가하고

두번째 Collector 객체에 두번째 PocketMonster 객체를 추가한다.

모든 Collector들의 토큰을 printAllTokens 메소드로 출력한다.

Scanner를 이용하여 tokenID 를 입력하고 (1 입력)

해당 토큰을 첫번째 Collector로부터 두번째 Collector 로 전송한다.

다시 한번 모든 Collector들의 토큰을 printAllTokens 메소드로 출력한다.

출력 예시:

```
Red(100):
name: Pikachu, id: 1, description: 1 Million Volt Electric Shock
Green(200):
name: GGobugi, id: 2, description: Surfing
----- Enter tokenId -----
1
This token is Pikachu
----- Successfully Transfered! -----
Red(100):
no pokemon
Green(200):
name: GGobugi, id: 2, description: Surfing
name: Pikachu, id: 1, description: 1 Million Volt Electric Shock
```

end
