



Lambda Expression 활용

학습내용

1 Java Collection Framework에서의 Lambda Expression의 활용

학습목표

- 기존의 프로그램에 Lambda Expression을 적용할 수 있다.
- 기존의 프로그램에 Lambda Expression을 적용함으로써 필요성을 설명할 수 있다.

Collection Framework에서의 Lambda Expression 활용

Class	Return Type	Method	Description
Iterable<E>	void	forEach(Consumer<? super E> action)	Iterable(Collection의 부모)의 각 instance에 대한 action 수행
Collection<E>	boolean	removeIf(Predicate<? super E> filter)	Collection의 각 instance에 대해 filter 조건이 성립하면 삭제
List<E>	boolean	replaceAll(UnaryOperator<E> operator)	List의 각 instance를 operator를 통해 교체
List<E>	boolean	sort(Comparator<? super E> c)	List를 비교방법 c에 의해 정렬
Map<K,V>	void	forEach(BiConsumer<? super K, ? super V> action)	Map의 각 key-value에 대한 action 수행
Map<K,V>	V	compute(K key, BiFunction<? super K, ? super V, ? extends V> remappingFunction)	Map의 key에 대한 key-value의 remapping을 수행함

Lambda Expression을 이용해
간략하게 프로그램을 작성 가능

Consumer<E>

Class	Return Type	Method	Description
Iterable <E>	void	forEach (Consumer <? super E> action)	Iterable(Collection의 부모)의 각 instance에 대한 action 수행

collection removelf

Class	Return Type	Method	Description
Collection ⟨E⟩	boolean	removelf (Predicate ⟨? super E⟩ filter)	Collection의 각 instance에 대해 filter 조건이 성립하면 삭제

replaceAll

Class	Return Type	Method	Description
List<E>	boolean	replaceAll (UnaryOperator<E> operator)	List의 각 instance를 operator를 통해 교체

replaceAll

UnaryOperator<E>

T를 받아 R을 반환하는 Functional
Interface에서 매개변수와 반환 값의 타입을
같은 타입으로 제한한 것

(각 instance) e를 받아 처리 후 반환

sort

Class	Return Type	Method	Description
List<E>	boolean	Sort (Comparator <? super E> c)	List를 비교방법 c에 의해 정렬

Map forEach

Class	Return Type	Method	Description
Map<K,V>	void	forEach (BiConsumer<? super K, ? super V> action)	Map의 각 key-value에 대한 action 수행

Map forEach

Class	Return Type	Method	Description
Map ⟨K,V⟩	V	Compute (K key, BiFunction ⟨? super K, ? super V, ? extends V⟩ remappingFunction)	Map의 key에 대한 key-value의 remapping을 수행함

그 외의 활용 : Thread

“Hello Thread”를 출력하는 독립적인 Thread 생성 후 시작

```
new Thread(new Runnable() {  
    @Override  
    public void run() {  
        System.out.println("Hello Thread");  
    }  
}).start();
```

Lambda Expression 적용

```
new Thread(() -> System.out.println("Hello Thread")).start();
```

그 외의 활용 : Java Swing

Java Swing JButton이 눌렸을 때의 이벤트 처리

```
new JButton("Button").addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        System.out.println("클릭됨");  
    }  
});
```

Lambda Expression 적용

```
new JButton("Button").addActionListener(e -> System.out.println("클릭됨"));
```

그 외의 활용 : Vertx-Web

웹 서버의 student 자원을 HTTP.GET으로 요청했을 때, 응답하는 핸들러 구현

```
Router.router(vertx).route("/:studentID").handler(new
    Handler<RoutingContext>() {
        @Override
        public void handle(RoutingContext ctx) {
            ctx.response().end("Hello " + ctx.pathParam("studentID"));
        }
    });
```

↓ Lambda Expression 적용

```
Router.router(vertx).route("/:studentID")
    .handler(ctx -> ctx.response().end("Hello " + ctx.pathParam("studentID")));
```

Remind

Java Collection Framework에서의
Lambda Expression 활용