**Lab3**

**Unit Test (tt)**

**By** [**traltb@fe.edu.vn**](mailto:traltb@fe.edu.vn)

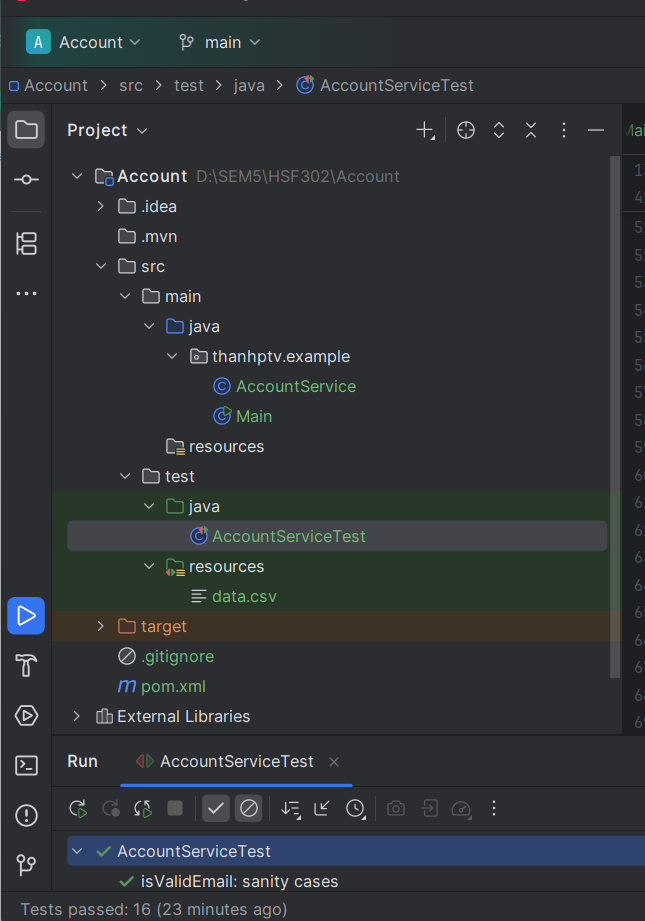
**Mục tiêu:**

* **Xây dựng Unit Test cho module**
* **Áp dụng file Unit Test, template của đồ án tốt nghiệp SEP490.**

**Họ tên: Phạm Thị Vân Thanh**

**MSSV: SE192641**

**CẤU TRÚC THƯ MỤC:**

****

**CLASS AccountService:**

package thanhptv.example;  
  
  
  
import java.util.regex.Pattern;  
  
public class AccountService {  
  
 // Regex email đơn giản – đủ cho lab (username@domain.tld)  
 private static final Pattern *EMAIL\_REGEX* =  
 Pattern.*compile*("^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,}$");  
  
 public boolean isValidEmail(String email) {  
 if (email == null) return false;  
 return *EMAIL\_REGEX*.matcher(email).matches();  
 }  
  
 */\*\*  
 \* Đăng ký hợp lệ khi:  
 \* - username không rỗng  
 \* - password dài > 6 ký tự  
 \* - email hợp lệ  
 \*/* public boolean registerAccount(String username, String password, String email) {  
 if (username == null || username.trim().isEmpty()) return false;  
 if (password == null || password.length() <= 6) return false;  
 return isValidEmail(email);  
 }  
}

**CLASS AccountServiceTest**

import static org.junit.jupiter.api.Assertions.\*;  
  
import java.util.stream.Stream;  
  
import org.junit.jupiter.api.DisplayName;  
import org.junit.jupiter.api.Test;  
import org.junit.jupiter.params.ParameterizedTest;  
import org.junit.jupiter.params.provider.MethodSource;  
import thanhptv.example.AccountService;  
  
class AccountServiceTest {  
  
 private final AccountService service = new AccountService();  
  
 */\*\* Bộ test cho isValidEmail – sanity check \*/* @Test  
 @DisplayName("isValidEmail: sanity cases")  
 void testIsValidEmail\_sanity() {  
 *assertTrue*(service.isValidEmail("john@example.com"));  
 *assertTrue*(service.isValidEmail("a.b-c+1@mail.co.uk"));  
 *assertFalse*(service.isValidEmail("no-at"));  
 *assertFalse*(service.isValidEmail("a@b"));  
 *assertFalse*(service.isValidEmail("user@domain."));  
 *assertFalse*(service.isValidEmail(null));  
 }  
  
 */\*\* Model dữ liệu \*/* static class Tc {  
 final String id, u, p, e; final boolean expected;  
 Tc(String id, String u, String p, String e, boolean expected) {  
 this.id = id; this.u = u; this.p = p; this.e = e; this.expected = expected;  
 }  
 @Override public String toString() {  
 return id + " | u=" + u + " | p=" + p + " | e=" + e + " => " + expected;  
 }  
 }  
  
 */\*\* 15 test case tương ứng UTCID01..UTCID15 \*/* static Stream<Tc> registerCases() {  
 return Stream.*of*(  
 new Tc("UTCID01","john123","pass123","john@example.com", true),  
 new Tc("UTCID02","", "pass123","john@example.com", false),  
 new Tc("UTCID03","alice", "123456", "alice@mail.com", false),  
 new Tc("UTCID04","bob", "1234567","b@c.com", true),  
 new Tc("UTCID05","bob", null, "b@c.com", false),  
 new Tc("UTCID06"," ", "pass1234","a@b.com", false),  
 new Tc("UTCID07","bob123", "password","bobmail.com", false),  
 new Tc("UTCID08","bob123", "password", null, false),  
 new Tc("UTCID09","user", "goodpass","first.last@sub.domain.co", true),  
 new Tc("UTCID10","user", "goodpass","user@domain.io", true),  
 new Tc("UTCID11","user", "goodpass","user@domain.", false),  
 new Tc("UTCID12",null, "goodpass","a@b.com", false),  
 new Tc("UTCID13","user", " 12 3456","u@d.com", true),  
 new Tc("UTCID14","user", "", "u@d.com", false),  
 new Tc("UTCID15","USER", "PassWord7","UPPER@EXAMPLE.COM", true)  
 );  
 }  
  
 @ParameterizedTest(name = "{0}")  
 @MethodSource("registerCases")  
 @DisplayName("registerAccount: 15 designed cases (UTCID01..UTCID15)")  
 void testRegisterAccount\_DesignedCases(Tc tc) {  
 boolean actual = service.registerAccount(tc.u, tc.p, tc.e);  
 *assertEquals*(tc.expected, actual, "Failed at " + tc.id);  
 }  
}

**KẾT QUẢ CHẠY TEST CASE**

**Ví dụ 1:**

