

Chuyên đề mở rộng New Date and Time

&

Annotations and Base64 Encoding

Mục tiêu

- ✓ Thực hành các lớp Datetime mới ở JDK 8
- ✓ Thực hành tạo các lớp hằng số Enum
- ✓ Sử dụng Annotation để chú thích mã nguồn Java
- ✓ Tùy chỉnh Annotation để bình luận, lưu ý, nhận xét phù hợp yêu cầu cụ thể.
- ✓ Sử dụng lớp Base64 mới để mã hóa và giải mã dữ liệu cơ bản, URL và MIME.

Bài thực hành số 1:

Yêu cầu: tạo lớp MainClass có hàm main(), khai báo các đối tượng Clock, Duration, LocalDate, LocalDateTime...và thực hành hiển thị, tính toán thời gian với các lớp mới này.

Code tham khảo:

```
Output - ChuyenDe.NewDateTimeAnnotation (run)

run:
clockUTC: SystemClock[Z]
clockDefaultZone: SystemClock[Asia/Bangkok]
1984 is leap year: true
No
Day of week: TUESDAY
Day of month: 23
Day of year: 297
Duration of 2 - 1 + 1 minute = 86460 (s)
Instant: 2018-04-19T10:25:50.652Z
Instant + 1 minute: 2018-04-19T10:26:50.652Z
LocalDateTime = 2018-04-19T17:25:50.676
LocalDateTime - 3 minutes = 2018-04-19T17:22:50.676
LocalDateTime + 1 year = 2019-04-19T17:25:50.676
LocalTime: 17:25:50.677
LocalTime + 15 seconds: 17:26:05.677
Cheer....: true
Age of me: 33
Month of me: 5
Day of me: 27
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
import java.time.Clock;
import java.time.Duration;
import java.time.Instant;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.LocalTime;
import java.time.MonthDay;
import java.time.Period;
```

```
/**
```

```
*
```

```
* @author minhvufo
```

```
*/
```

```
public class MainClass {
```

```
/**
```

```
* @param args the command line arguments
```

```
*/
```

```
public static void main(String[] args) {
```

```
// 1. Clock là lớp lấy thời gian mới trong JDK 8
```

```
Clock clockUTC = Clock.systemUTC();
```

```
System.out.println("clockUTC: " + clockUTC);
```

```
Clock clockDefaultZone = Clock.systemDefaultZone();
```

```
System.out.println("clockDefaultZone: " + clockDefaultZone);
```

```
// 2. LocalDate là lớp thao tác với ngày
```

```
LocalDate localDate = LocalDate.of(1984, 10, 23);
```

```
System.out.println("1984 is leap year: " + localDate.isLeapYear());
```

```
if (localDate.isAfter(LocalDate.now(clockUTC))) {
```

```

        System.out.println("Yes");
    } else {
        System.out.println("No");
    }

    System.out.println("Day of week: " + localDate.getDayOfWeek());
    System.out.println("Day of month: " + localDate.getDayOfMonth());
    System.out.println("Day of year: " + localDate.getDayOfYear());

    // 3. Duration sử dụng trong việc tính toán khoảng thời gian
    Duration dur = Duration.ofDays(2);
    System.out.println("Duration of 2 - 1 + 1 minute = " +
dur.minusDays(1).plusMinutes(1).getSeconds() + " (s)");

    // 4. Instant sử dụng với mục đích lấy/ thao tác thời gian GMT
    Instant ins = Instant.now();
    System.out.println("Instant: " + ins);
    System.out.println("Instant + 1 minute: " + ins.plusSeconds(60));

    // 5. LocalDateTime là lớp hỗ trợ tính toán với ngày giờ
    LocalDateTime localDateTime = LocalDateTime.now();
    System.out.println("LocalDateTime = " + localDateTime);
    System.out.println("LocalDateTime - 3 minutes = " + localDateTime.minusMinutes(3));
    System.out.println("LocalDateTime + 1 year = " + localDateTime.plusYears(1));

    // 6. LocalTime là lớp hỗ trợ xử lý về thời gian giờ
    LocalTime localTime = LocalTime.now();
    System.out.println("LocalTime: " + localTime);
    System.out.println("LocalTime + 15 seconds: " + localTime.plusSeconds(15));

    // 7. MonthDay dùng để so sánh, đối chiếu, tính toán ngày trong tháng
    MonthDay monthDay = MonthDay.now();
    MonthDay myBirthday = MonthDay.of(4, 19);
    System.out.println("Cheer....: " + monthDay.equals(myBirthday));

    // 8. Period có mục đích tính toán giai đoạn giữa các khoảng thời gian
    Period period = Period.between(localDate, LocalDate.now());
    System.out.println("Age of me: " + period.getYears());
    System.out.println("Month of me: " + period.getMonths());
    System.out.println("Day of me: " + period.getDays());
}
}

```

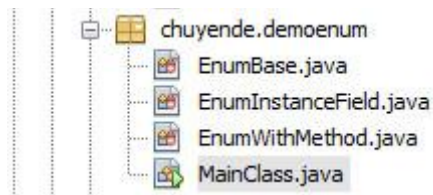
Bài thực hành số 2:

Yêu cầu: tạo 3 lớp Enum như sau:

1. Enum cơ bản
2. Enum có instance field

3. Enum có method

Tạo lớp có hàm main() và sử dụng các Enum trên



Code tham khảo:

File EnumBase.java

```
public enum EnumBase {  
    HOAT_DONG,  
    NGUNG,  
    TREO,  
    XOA  
}
```

File EnumInstanceField.java

```
public enum EnumInstanceField {  
    GOOGLE("www.google.com"),  
    YAHOO("www.yahoo.com"),  
    FACEBOOK("www.facebook.com");  
  
    String url;  
  
    private EnumInstanceField(String link) {  
        this.url = link;  
    }  
  
    public String getUrl() {  
        return url;  
    }  
}
```


File EnumWithMethod.java

```
public enum EnumWithMethod {  
    CONG,  
    TRU,  
    NHAN,  
    CHIA;  
}
```

```
double tinhToan(double a, double b) {  
    switch (this) {  
        case CONG:  
            return a + b;  
        case TRU:  
            return a - b;  
        default:  
            return 0;  
    }  
}
```

File MainClass.java

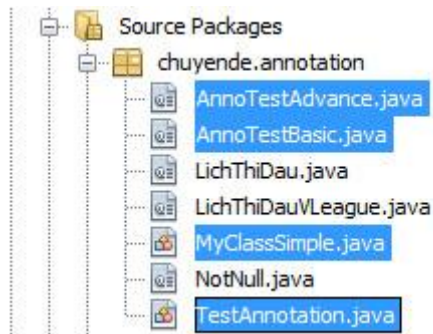
```
public class MainClass {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // Enum với hằng số cơ bản  
        System.out.println("Enum exam basic: " + EnumBase.HOAT_DONG);  
  
        // Enum với thể hiện biến  
        System.out.println("Enum with instance field: " +  
EnumInstanceField.FACEBOOK.getUrl());  
  
        // Enum với phương thức  
        System.out.println("Enum with method: " +  
EnumWithMethod.TRU.tinhToan(8, 5));  
    }  
}
```



```
Enum exam basic: HOAT_DONG  
Enum with instance field: www.facebook.com  
Enum with method: 3.0
```

Bài thực hành số 3:

Yêu cầu: tạo file annotation có tên như hình (bôi màu xanh), sử dụng nó để chú thích mã nguồn Java



Code tham khảo:

AnnoTestBasic.java

```
package chuyende.annotation;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

/**
 *
 * @author minhvuvc
 */
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface AnnoTestBasic {

    public boolean enable() default true;

    String createBy() default "MinhVT";
}
```

AnnoTestAdvance.java

```
package chuyende.annotation;

/**
 *
 * @author minhvuvc
 */
public @interface AnnoTestAdvance {

    String monthDay() default "1st";
}
```

MyClassSimple.java

`package` chuyende.annotation;

```
/**
 *
 * @author minhvuvc
 */
public class MyClassSimple {

    @Deprecated // <= Annotation build-in của Java
    void methodOne() {
        System.out.println("Todo something....");
    }

    void methodTwo() {
        System.out.println("Todo something....");
    }
}
```

TestAnnotation.java

`package` chuyende.annotation;

```
/**
 *
 * @author minhvuvc
 */
public class TestAnnotation extends MyClassSimple {

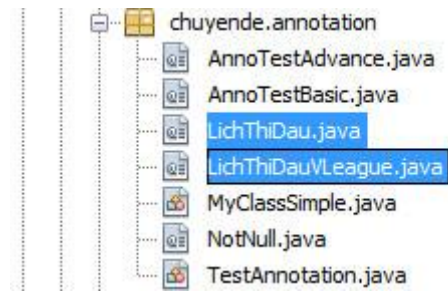
    void function1(@NotNull String svd) {

        // Test Annotation
        methodOne(); // <= Hiệu quả của Annotation @Deprecated
        test(null, 3);
    }

    @Override // <= Annotation build-in của Java
    void methodTwo() {
        super.methodTwo(); //To change body of generated methods, choose Tools |
        Templates.
    }
}
```

Bài thực hành số 4:

Yêu cầu: tạo annotation cho phép lặp



Code tham khảo:

LichThiDau.java

```
package chuyende.annotation;

import java.lang.annotation.Repeatable;

/**
 * @author minhvuvc
 */
@Repeatable(LichThiDauVLeague.class)
public @interface LichThiDau {

    String ngayTrongTuan();

    String ngayTrongThang();

    String dauTuan() default "T3-T5-T7";

    boolean sanNha() default true;

}
```

LichThiDauVLeague.java

```
package chuyende.annotation;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

/**
 *
 */
@Retention(RetentionPolicy.RUNTIME)
```



```
@Target(ElementType.METHOD)
public @interface LichThiDauVLeague {

    LichThiDau[] value();

}
```

TestAnnotation.java

```
package chuyende.annotation;

/**
 *
 * @author minhvufc
 */
public class TestAnnotation extends MyClassSimple {

    @LichThiDau(ngayTrongTuan = "CN", ngayTrongThang = "Last")
    @LichThiDau(ngayTrongTuan = "CN", ngayTrongThang = "Last", sanNha =
true)
    void function1(@NotNull String svd) {

        // Test Annotation
        methodOne(); // <= Hiệu quả của Annotation @Deprecated
        test(null, 3);
    }

    @Override // <= Annotation build-in của Java
    void methodTwo() {
        super.methodTwo(); //To change body of generated methods, choose Tools |
Templates.
    }

    /**
     * Cạn lời!
     *
     * @param param1
     * @param param2
     */
    @AnnoTestBasic(createBy = "MinhDaiCa", enable = false)
    @AnnoTestAdvance(monthDay = "FuckU")
    public void test(String param1, int param2) {
        System.out.println(param1 + param2);
    }
}
```

Bài thực hành số 5:

Yêu cầu: thực hành mã hóa và giải mã dữ liệu với Base64 mới trong JDK8

```
Output - Chuyển Đổi NewDate TimeAnnotation Data
```

```
RUN:  
1. Base64 cũ  
Base64Encode: RnVja2luZw==  
Base64.Encode: RnVja2luZw==  
  
2. Simple  
strEncode = Q29uZyBob2EgQCB4YSB0b2kgI2NodSAkbmdoaWEgJnZpZXRebmFt  
strDecode = Cong hoa @ xa hoi #chu $nghia &viet^nam  
  
3. URL  
encodedUrl: aHR0cHM6Ly93dGcuZ29vZ2xlLmNmVm56Lz9nZmVfcmQ9Y3ImZWk9ZHpiRlYmZ3dzX3JkPXNzbCkxPWphdmE=  
decodedUrl: https://www.google.co.nz/?gfe_rd=cr&ei=dzbFV6qws_rd=ssl&q=java  
  
4. MIME  
Base64 Encoded String (MIME) :V210aCBKYYZhIDgsIEPhu5luZyBIw7JhIFjDoySI4buZASBDaOG7pyBOZ2jEqWEgVmmhd4d0IE5h  
bs4gqmFzZY0IGhncyBmaWshbGx5IGdvcbCpdHMgZHVL1LiBKYYZhIDggbm9SIGhncyBpbmJlaWx0  
YWVuZ29kZXIgw5kiGRlY29kZXIgZm9yeIEJhc2U2NCBlbnV2GluZy4gSW4gSmF2YSA4LCB3BSbj  
YGgdXNllIHROcmVlIHRS5oGVZIG9mIEJhc2U2NCBlbnV2GluZy4=  
Base64 Decoded String (MIME) :With Java 8, Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam. Base64 has finally got its due. Java 8 now has inbuilt encoder and decoder for B  
  
5. Lưu và đọc dữ liệu ảnh khi MÃ HÓA và GIẢI MÃ theo Base64  
image = /9j/4AAQSkZJRgABAQEAASABIAAD/2wCEAAQDAwQDAwQEAWQFBAQFBgcHBsgYGBgOUJCggKDwOQEASNDw4RExgUERIXeg4PFRwVfxkZGxsEBQdHx0aHxgaGxoSBAUFBgUGDAcHDBorDxDxEaGhoaGhoaGhoa  
Copy xong  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Code tham khảo:

MainClass.java

```
package chuyende.base64;
```

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.nio.file.Files;
import java.util.Base64;
import java.util.logging.Level;
import java.util.logging.Logger;
import sun.misc.BASE64Encoder;
```

```
/**
 *
 * @author minhvuvc
 */
public class MainClass {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
```

```
// 1. Base64 cũ
System.out.println("1. Base64 cũ");
System.out.println("BASE64Encode: " + new
BASE64Encoder().encode("Fucking".getBytes()));
System.out.println("Base64.Encode: " +
Base64.getEncoder().encodeToString("Fucking".getBytes()));

// 2. Simle
System.out.println("\n\n2. Simle");
Base64.Encoder encoder = Base64.getEncoder();
Base64.Decoder decoder = Base64.getDecoder();
String strEncode = encoder.encodeToString(new String("Cong hoa @ xa hoi
#chu $nghia &viet^nam").getBytes());
System.out.println("strEncode = " + strEncode);
String strDecode = new String(decoder.decode(strEncode));
System.out.println("strDecode = " + strDecode);

// 3. URL
System.out.println("\n\n3. URL");
String originalUrl =
"https://www.google.co.nz/?gfe_rd=cr&ei=dzbFV&gws_rd=ssl#q=java";
String encodedUrl =
Base64.getUrlEncoder().encodeToString(originalUrl.getBytes());
byte[] decodedBytes = Base64.getUrlDecoder().decode(encodedUrl);
String decodedUrl = new String(decodedBytes);
System.out.println("encodedUrl: " + encodedUrl);
System.out.println("decodedUrl: " + decodedUrl);

// 4. MIME
System.out.println("\n\n4. MIME");
try {
    String str = "With Java 8, Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam. Base64
has finally got its due. "
        + "Java 8 now has inbuilt encoder and decoder for Base64 encoding. "
        + "In Java 8, we can use three types of Base64 encoding.";

    byte mimeBytes[] = str.getBytes("utf-8");
    String mimeEncodedString =
Base64.getMimeEncoder().encodeToString(mimeBytes);
    System.out.println("Base64 Encoded String (MIME) :" +
mimeEncodedString);

    String originalMIME = new
String(Base64.getMimeDecoder().decode(mimeEncodedString.getBytes()));
```

```
        System.out.println("Base64 Decoded String (MIME) :" + originalMIME);
    } catch (UnsupportedEncodingException ex) {
        Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null,
ex);
    }

    // 5. Lưu và đọc dữ liệu ảnh khi MÃ HÓA và GIẢI MÃ theo Base64
    System.out.println("\n\n5. Lưu và đọc dữ liệu ảnh khi MÃ HÓA và GIẢI MÃ
theo Base64");
    MainClass main = new MainClass();
    try {
        main.docAnh();
    } catch (IOException ex) {
        Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

/**
 * Thực hành mã hóa file ảnh bằng Base64 -> in ra, giải mã -> lưu lại
 *
 * @throws FileNotFoundException
 * @throws IOException
 */
private void docAnh() throws FileNotFoundException, IOException {
    String path = "D:/donut_cake.jpg";
    String new_path = "D:/donut_cake_new.jpg";
    byte arrByte[] = Files.readAllBytes(new File(path).toPath());

    Base64.Encoder encoder = Base64.getEncoder();
    String image = encoder.encodeToString(arrByte);

    System.out.println("image = " + image);

    // Copy ảnh mới
    Base64.Decoder decoder = Base64.getDecoder();
    byte newImage[] = decoder.decode(image);
    FileOutputStream fos = new FileOutputStream(new_path);
    fos.write(newImage);
    fos.flush();
    fos.close();
    System.out.println("Copy xong");
}
}
```