**AIM: Apply Prototype Design Pattern on the Domain Chosen**

**Domain**: **Cybersecurity**

**Usage:** Prototyping Email Parser (Deep Clone) and URL Analyzer (Shallow Clone)

|  |
| --- |
| class Sender {  // basic sender class made to see if any changes occur in EmailParser clone      private String sender;      public Sender(String sender) {          this.sender = sender;      }      public String getSender() {          return this.sender;      }      public void setSender(String value) {          this.sender = value;      }      public String toString() {          return "sender=" + sender;      }  }  class EmailParser implements Cloneable {      private String email;      private Sender sender;      private String subject;      private String content;      // ... other fields and methods      public EmailParser(String email, String sender, String subject, String content) {          this.email = email;          this.sender = new Sender(sender);          this.subject = subject;          this.content = content;      }      public Object clone() throws CloneNotSupportedException { // Deep Copy          try {              EmailParser clone = (EmailParser) super.clone();              clone.email = new String(this.email);              clone.sender = new Sender(this.sender.getSender());              clone.subject = new String(this.subject);              clone.content = new String(this.content);              return clone;          } catch (CloneNotSupportedException e) {              e.printStackTrace();              return null;          }      }      public String getEmail() {          return this.email;      }      public void setEmail(String email) {          this.email = email;      }      public String getSender() {          return this.sender.getSender();      }      public void setSender(String value) {          this.sender.setSender(value);      }      public String getSubject() {          return this.subject;      }      public void setSubject(String value) {          this.subject = value;      }      public String getContent() {          return this.content;      }      public void setContent(String value) {          this.content = value;      }      public String toString() {          return "EmailParser [email=" + email + ", sender=" + sender + ", subject=" + subject + ", content=" + content + "]";      }  }  class Domain { // basic Domain class made to see if any changes occur in URLAnalyzer clone      private String domain;      public Domain(String domain) {          this.domain = domain;      }      public String getDomain() {          return this.domain;      }      public void setDomain(String value) {          this.domain = value;      }      public String toString() {          return "domain=" + domain;      }  }  class URLAnalyzer implements Cloneable {      private String url;      private Domain urlDomain;      public URLAnalyzer(String url, String domain) {          this.url = url;          urlDomain = new Domain(domain);      }      public String getURL() {          return this.url;      }      public void setURL(String value) {          this.url = value;      }      public String getDomain() {          return urlDomain.getDomain();      }      public void setDomain(String value) {          urlDomain.setDomain(value);      }      public URLAnalyzer clone() throws CloneNotSupportedException {          try {              return (URLAnalyzer) super.clone();          } catch (CloneNotSupportedException e) {              e.printStackTrace();              return null;          }      }      public String toString() {          return "URLAnalyzer [url=" + url + ", " + urlDomain + "]";      }  }  public class prototypeDesign {      public static void main(String[] args) throws CloneNotSupportedException {          EmailParser email = new EmailParser("example@email.com", "John Doe", "Important message",                  "This is a phishing email.");          EmailParser emailClone = (EmailParser) email.clone();          URLAnalyzer url = new URLAnalyzer("http://example.com", "example.com");          URLAnalyzer urlClone = url.clone();          // Changes          emailClone.setSender("Jane Doe");          url.setDomain("pdeu.ac.in");          System.out.println("Deep Copy");          System.out.println(email);          System.out.println(emailClone);          System.out.println("Shallow Copy");          System.out.println(url);          System.out.println(urlClone);      }  } |

**Output:**

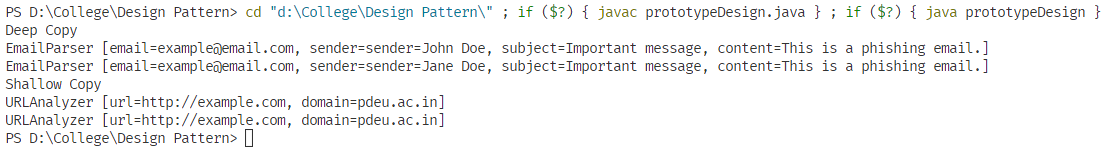
****

Figure 1: Output of the above program