

// Bill.java

```
© CitationFormatter

strategy: CitationStrategy

setStrategy(strategy: CitationStrategy): void
formatName(firstName: String, lastName: String): String

© CitationStrategy

oformatName(firstName: String, lastName: String): String

© ACMFormatStrategy

oformatName(firstName: String): String

oformatName(firstName: String): String

oformatName(firstName: String, lastName: String): String
```

```
public abstract class Bill {
    protected int minutes;
    protected double rate;
    protected static final double GST = 0.9;
    public Bill(int m, double r) {
         minutes = m;
         rate = r;
    // Template method
    public final double getBillableAmount() {
         double base = calculateBase();
         double tax = calculateTax(base):
         return base + tax;
    // Abstract methods for the steps that vary
    protected abstract double calculateBase();
    protected abstract double calculateTax(double base);
public class PasswordValidator {
   Run | Debug
   public static void main(String[] args) {
          Scanner scanner = new Scanner(System.in);
          System.out.print(s:"Enter your password: ");
          String password = scanner.nextLine();
          if (isValidPassword(password)) {
              System.out.println(x:"Password is valid.");
          } else {
              System.out.println(x:"Password is invalid.");
          System.out.println("An error occurred: " + e.getMessage());
   private static boolean isValidPassword(String password) {
       String regex = "^{?=.*[a-z])(?=.*[A-Z])(?=.*[!@#\\$%]).{12,}$";
       Pattern pattern = Pattern.compile(regex);
      Matcher matcher = pattern.matcher(password);
       return matcher.matches();
    public static void main(String[] args) {
        //"Hello World"
        String string = new String(original: "Hello");
        String string2 = string.concat(str:"_World");
        System.out.println(string2);
        String string3 = string2.replace(target:"_", replacement:" ");
        System.out.println(string3);
        StringBuffer buffer = new StringBuffer(str:"Hello");
        buffer.append(str:"_World");
        System.out.println(buffer);
        buffer.setCharAt(index:5, ch:' ');
        System.out.println(buffer);
```

```
public class Prof {
    private String name;
                   (C) MyStack
                                                       private Date midtermDate;
private ArrayList<MidtermHandler> midtermHandlers;
      list : List<Object>
                                                       public Prof(String aName) {
      • push(element : Object) : void
                                                            this.name = aName;
this.midtermHandlers = new ArrayList<MidtermHandler>();
      o pop(): Object
                                                       public Date getMidterm() {
                                                            return this.midtermDate;
                           delegates
                                                       public String getName() {
    return this.name;
                      (I) List
                                                       public void setMidterm(Date date, Prof prof) {
    this.midtermDate = date;
add(index : int, element : Object) :
                                                            CourseEvent event = new CourseEvent(this.midtermDate, prof);
for(MidtermHandler s: this.midtermHandlers){
remove(index : int) : Object
o isEmpty(): boolean
                                                                  s.handleMidterm(event);
                                                       public void postponeMidterm(Date date, Prof prof){
                                                            this.midtermDate = date;
                                                            CourseEvent event = new CourseEvent(date, prof);
for(MidtermHandler s: this.midtermHandlers){
                    (C) ArrayList
                                                                  s.handlePostponeMidterm(event);
                                                       public void addMidtermHandler(MidtermHandler mh){
                                                            this.midtermHandlers.add(mh);
                name: String }
midtermDate: Date
midtermHandlers: ArrayList<MidtermHandler>

    Prof(String)
    getMidterm(): Date

    getName(): String
    setMidterm(Date, Prof)

                postponeMidterm(Date, Prof)
                 addMidtermHandler(MidtermHandler)
                   CourseEvent
                                                           MidtermHandler
             midtermDate: Date

    CourseEvent(Date, Prof)
    getMidtermDate(): Date
    getProf(): Prof

                                               handleMidterm(CourseEvent)
                                                handlePostponeMidterm(CourseEvent)
                                                            1
                                                       , împlements
                                                                              implements
                                      Student
                                                                                  TeachingAssistant
                      name: String
midterm: Date
                                                                      name: String
midterm: Date

    TeachingAssistant(String)
    getName(): String
    handleMidterm(CourseEvent)
    handlePostponeMidterm(CourseEvent)
                      Student(String)
                       getName(): String
handleMidterm(CourseEvent)

    handlePostponeMidterm(CourseEvent)

          //IEEEFormatStrategyTest.java
          public class IEEEFormatStrategyTest {
                @Test
                public void testFormatName(){
                      CitationStrategy strategy = new IEEEFormatStrategy();
                      assertEquals("B. Esfandiari", strategy.formatName("Babak", "Esfandiari"));
```

```
// TodoController.java
                                                                                             // Main.java
public class TodoController
                                                                                             public class Main {
   private TodoComposite todoList;
                                                                                                 Run | Debug
    private TodoView view;
                                                                                                 public static void main(String[] args) {
                                                                                                      TodoProject mainProject = new TodoProject(name: "Main Project");
    public TodoController(TodoComposite todoList, TodoView view) {
       this.todoList = todoList:
                                                                                                      TodoItem item1 = new TodoItem(text:"Complete Java assignment");
       this.view = view;
                                                                                                      TodoItem item2 = new TodoItem(text: "Read design patterns book");
                                                                                                      mainProject.add(item1);
   public void saveTodoListToFile(String filename) {
       try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(filename))) {
                                                                                                      mainProject.add(item2);
           out.writeObject(todoList);
       } catch (IOException e) {
                                                                                                      TodoProject subProject = new TodoProject(name: "Sub Project");
           e.printStackTrace():
                                                                                                      subProject.add(new TodoItem(text:"Sub task 1"));
                                                                                                      subProject.add(new TodoItem(text:"Sub task 2"));
   public void loadTodoListFromFile(String filename) {
                                                                                                      mainProject.add(subProject);
       try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(filename))) {
           todoList = (TodoComposite) in.readObject();
                                                                                                      TodoView view = new TodoView();
       } catch (IOException | ClassNotFoundException e) {
           e.printStackTrace():
                                                                                                      TodoController controller = new TodoController(mainProject, view);
                                                                                                      controller.displayTodoList();
   public void addTodoItem(TodoComposite item) {
                                                                                                      controller.displayXMLTodoList();
       todoList.add(item);
                                                                                                      controller.saveTodoListToFile(filename:"todolist.ser");
                                                                                                      controller.loadTodoListFromFile(filename:"todolist.ser");
   public void displayTodoList() {
       view.displayTodoList(todoList);
   }
   public void displayXMLTodoList() {
       view.displayXMLTodoList(todoList);
                                                                                                        // TodoView.java
                                                                                                        public class TodoView {
                                                                                                             public void displayTodoList(TodoComposite todoList) {
    //TodoProject.java
    public class TodoProject extends TodoComposite {
                                                                                                                  todoList.display();
        private String name;
                                                     this class is associated with ---- this class
        public TodoProject(String name) {
                                                     this class is dependent upon - - - > this class
                                                                                                             public void displayXMLTodoList(TodoComposite todoList) {
            this.name = name;
                                                                                                                  todoList.toXML();
                                                          this class inherits from -
                                                                                      this class
        @Override
                                                                  this class has -
                                                                                      this interface
        public void display() {
            System.out.println(name + ":");
                                                      this class is a realisation of - - - > this class
                                                                                                                  // TodoComposite.iava
            for (TodoComposite item : children) {
                                                                                                                  public abstract class TodoComposite implements Serializable {
               item.display();
                                                     in navigate from this class to ---
                                                                                      this class
                                                                                                                      protected List<TodoComposite> children = new ArrayList<>();
                                                                                                                      public void add(TodoComposite item) {
        public String toXML() {
                                                                                                                          children.add(item);
            StringBuilder xmlBuilder = new StringBuilder(str:"<todoProject>\n");
                                                                                        MVC Todolist
            for (TodoComposite item : children)
               xmlBuilder.append(item.toXML()).append(str:"\n");
                                                                                                                      public void remove(TodoComposite item) {
                                                                                      Composite, XML,
                                                                                                                          children.remove(item):
            xmlBuilder.append(str:"</todoProject>");
                                                                                         Serialization
            System.out.println(xmlBuilder.toString());
            return xmlBuilder.toString();
                                                                                                                      public abstract void display();
                                                                                                                      public abstract String toXML();
                                                                             C TodoController
                                                            □ todoList : TodoComposite
                                                                                                                    //TodoItem.iava
    C Todoltem
                                 C TodoProject
                                                            □ view : TodoView
                                                                                                                    public class TodoItem extends TodoComposite {
   text : String
                                 name : String
                                                            • TodoController(todoList : TodoComposite, view : TodoView)
                                                                                                                         private String text;
                                 display(): void

    addTodoltem(item : TodoComposite) : void

   display(): void
   toXML(): String
                                toXML(): String
                                                            displayTodoList(): void
                                                                                                                         public TodoItem(String text) {

    saveTodoListToFile(filename : String) : void

                                                            • loadTodoListFromFile(filename : String) : void
                                                                                                                             this.text = text;
                                               extends
                                                                                                                         public void display() {
                                            A TodoComposite
                                                                                                                             System.out.println("- " + text);
                                     children : List<TodoComposite>
                                                                                       C Todo√iew
         implements
                         implements

    add(item : TodoComposite) : void

                                     remove(item : TodoComposite) : void

    displayTodoList(todoList : TodoComposite) : void

                                                                                                                         public String toXML() {
                                     display(): void
                                                                                                                             return "<todo>" + this.text + "</todo>";

    toXML(): String

                                    .
implements
                     «Interface»
                 Serializable
```