МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №6

Специальность ПО9(з)

Выполнил
Д. Н. Кухарев,
студент группы ПО9

Проверил
А. А. Крощенко,
ст. преп. кафедры ИИТ,
«___k_________2024 г.

Брест 2024

Цель работы: приобрести навыки применения паттернов проектирования при решении практических задач с использованием языка Java.

Вариант 9

Задание:

• Прочитать задания, взятые из каждой группы.

• Определить паттерн проектирования, который может использоваться при реализации задания.

Пояснить свой выбор.

• Реализовать фрагмент программной системы, используя выбранный паттерн. Реализовать все необходимые дополнительные классы.

Задание 1. Проект «Бургер-закусочная». Реализовать возможность формирования заказа из определенных позиций (тип бургера (веганский, куриный и т.д.)), напиток (холодный – пепси, кока-кола и т.д.; горячий – кофе, чай и т.д.), тип упаковки – с собой, на месте. Должна формироваться итоговая стоимость заказа.

Выполнение:

И так, у нас есть фиксированная комплектация заказа на 3 предмета: бургер, напиток, упаковка. Разнообразные бургеры представлены одним классом: бургера, но имеют разную реализацию. В таком случае используется паттерн проектирования «Стратегия».

Код программы

Main.java, метод formOrder():

```
public static void formOrder(){
    Drink[] drinks;
    Scanner read = new Scanner(System.in);
    String action;
    Burger burger;
    Drink drink;
    Pack pack = null;
    System.out.print("Select burger number: ");
    action = read.next();

    int action_num = 0;
    try{
      action_num = Integer.parseInt(action)-1;
    }catch(Exception ex){
      System.out.println("Please enter correct number");
      return;
    }
    if(action_num >= burgers.length){
      System.out.println("There's no " + action + "th burger");
      return;
    }else{
      burger = burgers[action_num];
    }

    System.out.println("Do you want to order hot drink? (N/y)");
    action = read.next();
    if(action.toLowerCase().equals("y")){
      drinks = hotDrinks;
      showHotDrinkList();
    }else{
            drinks = coldDrinks;
            showColdDrinkList();
        }

        System.out.print("Select drink number: ");
        action = read.next();
        try{
            action_num = Integer.parseInt(action)-1;
        }catch(Exception ex){
            System.out.println("Please enter correct number");
            return;
        }
        if(action_num >= drinks.length){
            System.out.println("There's no " + action + "th drink");
            return;
        }else{
            drink = drinks[action_num];
        }
        System.out.println("Would you like to take your order with you? (N/y)");
        action = read.next();
        if(action.toLowerCase().equals("y")){
            pack = new ToGo();
        }else{
            pack = new OnSite();
        }
        BurgerDiner.MakeOrder(burger, drink, pack);
    }
```

Burger.java:

```
package Burger;

public abstract class Burger {
    final public static int VEGAN_BURGER_COST = 6;
    final public static int HAMBURGER_COST = 8;
    final public static int CHEESEBURGER_COST = 10;
    final public static int CHICKEN_BURGER_COST = 9;
    final public static String VEGAN_COMPOSITION = "Bun, soy cutlet, tofu, onion, tomatoes, lettuce leaves, soy
based sauce";
    final public static String CHICKEN_COMPOSITION = "Bun, fried chicken fillet, onion, lettuce leaves, special
sauce";
    final public static String CHEESE_COMPOSITION = "Bun, cutlet, cheese, onion, lettuce leaves, special sauce";
    final public static String HAM_COMPOSITION = "Bun, ham slice, marinated onion, lettuce leaves, special sauce";
    CookStrategy cookStrategy;
    String name;
    public int buy(){
        return cookStrategy.choose();
    }
    public abstract void showInfo();
    protected static void show(String name, String composition, int price){
        System.out.println(name + "\n\t - Composition: " + composition + "\n\t - Price: $" + price);
    }
}
```

CookStrategy.java:

```
package Burger;

public interface CookStrategy {
    public int choose();
}
```

CheeseBurger.java:

```
package Burger;
```

```
import Burger.Burger;

public class CheeseBurger extends Burger {
   public CheeseBurger(){
      this.name = "Cheeseburger";
      this.cookStrategy = new CookCheeseBurger();
   }
   public void showInfo(){
      show(name, Burger.CHEESE_COMPOSITION, Burger.CHEESEBURGER_COST);
   }
}
```

Drink.java:

```
package Drink;

public abstract class Drink {
   protected String name;
   DrinkStrategy drinkStrategy;
   public double selectDrink(){
      return drinkStrategy.select();
   }
   public abstract void showInfo();
   protected static void show(String name, double price){
      System.out.println(name + "\n\t - Price: $" + price);
   }
}
```

DrinkStrategy.java:

```
package Drink;

public interface DrinkStrategy {
   public double select();
}
```

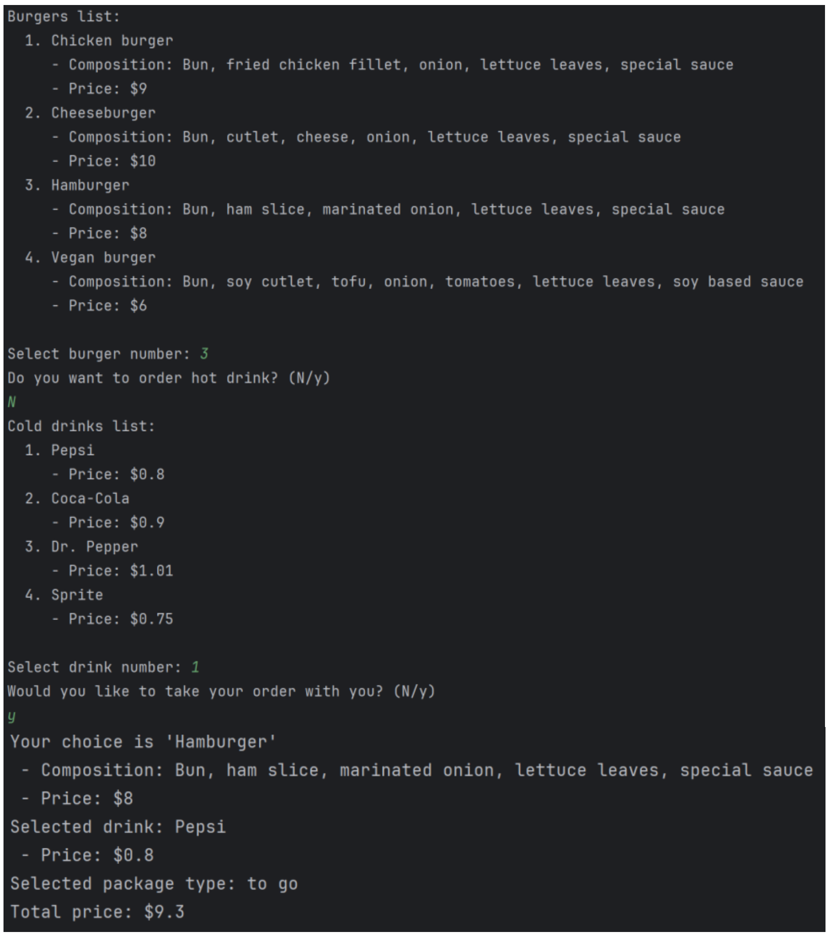
Cold.java:

```
package Drink;

public abstract class Cold extends Drink{
   final public static String PEPSI = "Pepsi";
   final public static double PEPSI_PRICE = 0.8;
   final public static String COLA = "Coca-Cola";
   final public static double COLA_PRICE = 0.9;
   final public static String SPRITE = "Sprite";
   final public static double SPRITE_PRICE = 0.75;
   final public static String DRPEPPER = "Dr. Pepper";
   final public static double DRPEPPER_PRICE = 1.01;
   public abstract void showInfo();
}
```

Pepsi.java:

```
package Drink;

public class Pepsi extends Cold{
   public Pepsi(){
      this.name = Cold.PEPSI;
      this.drinkStrategy = new PepsiStrategy();
   }
   public void showInfo(){
      show(name, Cold.PEPSI_PRICE);
   }
}
```

Рисунки с результатами работы программы



```
Burgers list:
  1. Chicken burger
     - Composition: Bun, fried chicken fillet, onion, lettuce leaves, special sauce
     - Price: $9
  2. Cheeseburger
     - Composition: Bun, cutlet, cheese, onion, lettuce leaves, special sauce
     - Price: $10
  3. Hamburger
     - Composition: Bun, ham slice, marinated onion, lettuce leaves, special sauce
     - Price: $8
  4. Vegan burger
     - Composition: Bun, soy cutlet, tofu, onion, tomatoes, lettuce leaves, soy based sauce
     - Price: $6

Select burger number: 3
Do you want to order hot drink? (N/y)
N
Cold drinks list:
  1. Pepsi
     - Price: $0.8
  2. Coca-Cola
     - Price: $0.9
  3. Dr. Pepper
     - Price: $1.01
  4. Sprite
     - Price: $0.75

Select drink number: 1
Would you like to take your order with you? (N/y)
y
Your choice is 'Hamburger'
 - Composition: Bun, ham slice, marinated onion, lettuce leaves, special sauce
 - Price: $8
Selected drink: Pepsi
 - Price: $0.8
Selected package type: to go
Total price: $9.3
```

Задание 2. Проект «Часы». В проекте должен быть реализован класс, который дает возможность пользоваться часами со стрелками так же, как и цифровыми часами. В классе «Часы со стрелками» хранятся повороты стрелок.

Выполнение:

Есть часы со стрелками, чтобы пользоваться ими как цифровыми можем применить адаптер, который будет переводить движения механических частей в электронное время.

Код программы

ArrawClock.java:

```
    public interface ArrowClock {
        public void showTime();
        public void setClocks(double rotation);
}
```

DigitalClock.java:

```
    public interface DigitalClock {
        public void showTime();
        public void setClocks(int hours, int minutes, int seconds);
}
```

ClockWithArrow.java:

```
    public class ClockWithArrow implements ArrowClock{
        final private int degreeAmount = 360;
        final private int nextCircleTransition = -1;
        final private int hoursToDegrees = 30;
        final private int minutesSecondsToDegrees = 6;
        final private int rotationsPerHour = 6;
        final private int minutesPerRotation = 10;
        final private int secondsPerRotation = 600;
        private int hourArrDegree;
        private int minuteArrDegree;
        private int secondArrDegree;
        ClockWithArrow(){
           hourArrDegree = 0;
           minuteArrDegree = 0;
           secondArrDegree = 0;
        }
        ClockWithArrow(double rotationAmount){
           hourArrDegree = (int)(rotationAmount/rotationsPerHour*hoursToDegrees);
           while (hourArrDegree > degreeAmount+nextCircleTransition){
              hourArrDegree -= degreeAmount;
           }
           minuteArrDegree = (int)(rotationAmount*minutesPerRotation*minutesSecondsToDegrees);
           while (minuteArrDegree > degreeAmount+nextCircleTransition){
              minuteArrDegree -= degreeAmount;
           }
           secondArrDegree = (int)(rotationAmount*secondsPerRotation*minutesSecondsToDegrees);
           while (secondArrDegree > degreeAmount+nextCircleTransition){
              secondArrDegree -= degreeAmount;
           }
        }
        @Override
        public void showTime() {
           System.out.printf("*Часы показывают*\nЧасы: %d\nМинуты: %d\nМинуты: %d\n\n",
     hourArrDegree/hoursToDegrees, minuteArrDegree/minutesSecondsToDegrees,
     secondArrDegree/minutesSecondsToDegrees);
        }
        @Override
        public void setClocks(double rotationAmount){
           System.out.println("Крутим-вертим устанавливаем время");
           hourArrDegree = (int)(rotationAmount/rotationsPerHour*hoursToDegrees);
           while (hourArrDegree > degreeAmount+nextCircleTransition){
              hourArrDegree -= degreeAmount;
           }
           minuteArrDegree = (int)(rotationAmount*minutesPerRotation*minutesSecondsToDegrees);
           while (minuteArrDegree > degreeAmount+nextCircleTransition){
              minuteArrDegree -= degreeAmount;
           }
           secondArrDegree = (int)(rotationAmount*secondsPerRotation*minutesSecondsToDegrees);
           while (secondArrDegree > degreeAmount+nextCircleTransition){
              secondArrDegree -= degreeAmount;
           }
   }
}
```

ClocksDigital.java:

```
public class ClocksDigital implements DigitalClock{
   int hours;
   int minutes;
   int seconds;
   final private int max_hours = 11;
   final private int max_minutes = 59;
   final private int max_seconds = 59;
   final private int min_time = 0;
   ClocksDigital(){
        hours = 0;
        minutes = 0;
        seconds = 0;
    }
    ClocksDigital(int hours, int minutes, int seconds){
        if((hours > max_hours || hours < min_time)
                || (minutes > max_minutes|| minutes < min_time)
                || (seconds > max_seconds || seconds < min_time)){
            throw new IllegalArgumentException("Wrong time!");
        }
        this.hours = hours;
        this.minutes = minutes;
        this.seconds = seconds;
    }
    @Override
    public void showTime() {
        System.out.printf("Time: %d:%d:%d\n", hours, minutes, seconds);
    }

    @Override
    public void setClocks(int hours, int minutes, int seconds) {
        System.out.println("Digital time clock time set");
        if((hours > max_hours || hours < min_time)
                || (minutes > max_minutes || minutes < min_time)
                || (seconds > max_seconds || seconds < min_time)){
            System.out.println("Wrong time!");
            return;
        }
        this.hours = hours;
        this.minutes = minutes;
        this.seconds = seconds;
    }
}
```

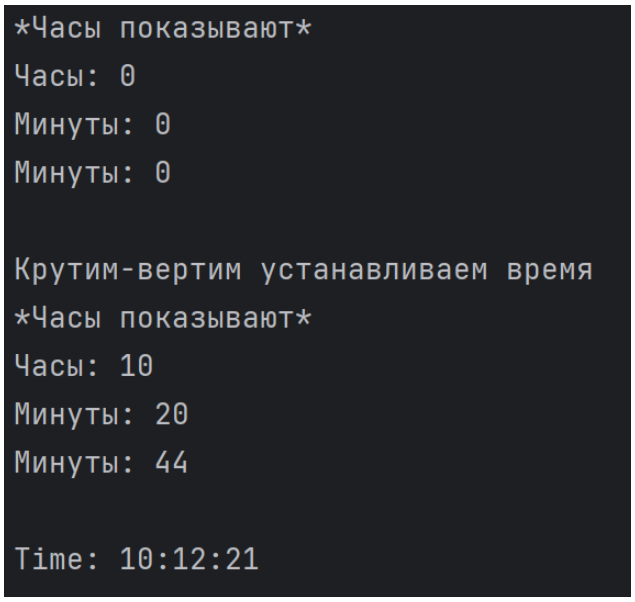
ArrowToDigitalAdapter.java:

```
    public class ArrowToDigitalAdapter implements DigitalClock{
        final private int minutesInHour= 60;
        final private int crownRatioToMinutes = 10;
        final private int crownRatioToSeconds = 600;
        ClockWithArrow arrowClock;
        ArrowToDigitalAdapter(ClockWithArrow arrowClock){
            this.arrowClock = arrowClock;
        }
        @Override
        public void showTime() {
            arrowClock.showTime();
        }

        @Override
        public void setClocks(int hours, int minutes, int seconds) {
            arrowClock.setClocks((((double) hours*minutesInHour/crownRatioToMinutes)
                +((double) minutes/crownRatioToMinutes)//1 crown rotation equals 10 minutes
                +((double) seconds/crownRatioToSeconds)));
        }
}
```

Main.java:

```
public class Main {

    public static void main(String[] args) {
        ClockWithArrow n = new ClockWithArrow();
        DigitalClock adapted = new ArrowToDigitalAdapter(n);
        adapted.showTime();
        adapted.setClocks(10, 20, 44);
        adapted.showTime();

        DigitalClock digital = new ClocksDigital(10, 12, 21);
        digital.showTime();
    }


}
```

Рисунки с результатами работы программы



```
*Часы показывают*
Часы: 0
Минуты: 0
Минуты: 0

Крутим-вертим устанавливаем время
*Часы показывают*
Часы: 10
Минуты: 20
Минуты: 44

Time: 10:12:21
```

Задание 3. Шифрование текстового файла. Реализовать класс-шифровщик текстового файла с поддержкой различных алгоритмов шифрования. Возможные варианты шифрования: удаление всех гласных букв из текста, изменение букв текста на буквы, получаемые фиксированным сдвигом из алфавита (например, шифром буквы а будет являться буква д для сдвига 4 и т.д.), применение операции исключающее или с заданным ключом.

Выполнение:

Снова есть одна задача и несколько её реализаций, нам подходит стратегия.

Код программы

Encryption.java:

```
package kdn.lab6.task1.encryption;
import org.apache.log4j.Logger;
/**
 * Parent class kdn.lab6.task1.encryption.Encryption for all encryption and decryption methods
(VowelsDelete, XOR, Atbash)
```

```
 */public class Encryption {
    String encryptedName;
    String toEncrypt;
    EncryptStrategy encryptStrategy;
    final static protected String TXT = ".txt";
    final static protected String ERROR_MESSAGE = "Wrong file type!";
    final static protected int NOT_FOUND = 0;

    /**
     * A common method for all kdn.lab6.task1.encryption.Encryption classes that performs direct encryption
     * @param fileToWrite
     * @return
     */
    public void encrypt(String fileToWrite){
        IFile.writeFile(encryptStrategy.encrypt(), fileToWrite);
    }
    /**
     * A common method for all kdn.lab6.task1.encryption.Encryption classes that performs direct decryption
     * @param fileToWrite
     * @return
     */
    public void decrypt(String fileToWrite){
        IFile.writeFile(encryptStrategy.decrypt(), fileToWrite);
    }
}
```

VowelsDelete.java:

```
/**
 * kdn.lab6.task1.encryption.VowelsDelete implements kdn.lab6.task1.encryption.Encryption by removing
vowels(eng, rus) from a text
 */
public class VowelsDelete extends Encryption {
    private static Logger logger = Logger.getLogger(VowelsDelete.class);
    /**
     * Creating a new object with the encryption method <code>'VowelsDelete'</code>,
     * that removes all vowels from a text file.
     * <br/>
     * Use {@link Encryption#encrypt(String fileToWrite)} for encryption.
     * <br/>
     * And {@link Encryption#decrypt(String fileToWrite)} for decryption.
     * @param pathToInitialFile the path to the file
     */
    public VowelsDelete(String pathToInitialFile){
        if(pathToInitialFile.indexOf(TXT) < NOT_FOUND){
            JOptionPane.showMessageDialog(null, ERROR_MESSAGE);
            throw new IllegalArgumentException(ERROR_MESSAGE);
        }
        toEncrypt = IFile.readFile(pathToInitialFile);
        encryptStrategy = new VowelsStrategy(toEncrypt);
        logger.info("New Vowels Delete object was created: " + this.toString());
    }
}
```

AtbashEncryption.java:

```
/**
 * kdn.lab6.task1.encryption.AtbashEncryption implements kdn.lab6.task1.encryption.Encrypti on by making
Atbash (fixed shift) encryption of a text
 */
public class AtbashEncryption extends Encryption {
    private static Logger logger = Logger.getLogger(AtbashEncryption.class);
    /**
     * Creating a new object with the encryption method <code>'AtbashEncryption'</code>,
     * that uses fixed ASCII alphabet shift to encrypt text
     * <br/>
     * Use {@link Encryption#encrypt(String fileToWrite)} for encryption.
     * <br/>
     * And {@link Encryption#decrypt(String fileToWrite)} for decryption.
     * @param pathToInitialFile
     * @param bias
     */
    public AtbashEncryption(String pathToInitialFile, int bias){
        if(pathToInitialFile.indexOf(Encryption.TXT) < Encryption.NOT_FOUND){
            JOptionPane.showMessageDialog(null, Encryption.ERROR_MESSAGE);
            throw new IllegalArgumentException(Encryption.ERROR_MESSAGE);
        }
        toEncrypt = IFile.readFile(pathToInitialFile);
        encryptStrategy = new AtbashStrategy(toEncrypt, bias);
        logger.info("New Atbash kdn.lab6.task1.encryption.Encryption object was created: " + this.toString());
    }
    public void setBias(int bias){
        encryptStrategy = new AtbashStrategy(toEncrypt, bias);
        logger.info("New bias for Atbash kdn.lab6.task1.encryption.Encryption object: " + this.toString());
    }
}
```

XorEncryption.java:

```
/**
 * kdn.lab6.task1.encryption.XorEncryption implements kdn.lab6.task1.encryption.Encryption by making Xor
encryption of a text using key
 */
public class XorEncryption extends Encryption {
    private static Logger logger = Logger.getLogger(XorEncryption.class);

    /**
     * Creating a new object with the encryption method <code>'XorEncryption'</code>,
     * that uses XOR operation for Input File and Key
     * <br/>
     * Use {@link Encryption#encrypt(String fileToWrite)} for encryption.
     * <br/>
     * And {@link Encryption#decrypt(String fileToWrite)} for decryption.
     * @param pathToInitialFile
     * @param key
     */
    public XorEncryption(String pathToInitialFile, String key){
        if(pathToInitialFile.indexOf(TXT) < NOT_FOUND){
            JOptionPane.showMessageDialog(null, ERROR_MESSAGE);
            throw new IllegalArgumentException(ERROR_MESSAGE);
        }
        toEncrypt = IFile.readFile(pathToInitialFile);
        encryptStrategy = new XorStrategy(toEncrypt, key);
        logger.info("New Xor kdn.lab6.task1.encryption.Encryption object was created: " + this.toString());
    }
    public void setKey(String key){
        encryptStrategy = new XorStrategy(toEncrypt, key);
        logger.info("New key Xor kdn.lab6.task1.encryption.Encryption object: " + this.toString());
    }
}
```

EncryptionStrategy.java:

```
package kdn.lab6.task1.encryption;
```

```
/**
 * Basic interface for all encryption strategies
 */
public interface EncryptStrategy {

   public String encrypt();
   public abstract String decrypt();
}
```

XorStrategy.java:

```
class XorStrategy implements EncryptStrategy {
   private static Logger logger = Logger.getLogger(XorStrategy.class);
   String toEncrypt, toDecrypt, key;

   XorStrategy(String toEncrypt, String key) {
      this.toEncrypt = toEncrypt;
      this.key = key;
   }

   @Override
   public String encrypt() {
      toEncrypt = xor(toEncrypt, key);
      logger.info("File was encrypted with Xor method");
      return toEncrypt;
   }

   @Override
   public String decrypt() {
      toDecrypt = xor(toEncrypt, key);
      logger.info("File was decrypted with Xor method");
      return toDecrypt;
   }

   public static String xor(String str, String key) {
      int str_length = str.length();
      int key_length = key.length();
      if(str_length < key_length){
         key = key.substring(0, str_length);
      }
      byte[] str_bytes = str.getBytes(StandardCharsets.UTF_8);
      byte[] key_bytes = key.getBytes(StandardCharsets.UTF_8);
      byte[] res_bytes = str_bytes;
      for (int i = 0, j = 0; i < str_bytes.length; ++i, ++j) {
         if(j > key_length-1){
            j = 0;
         }
         res_bytes[i] = (byte) (str_bytes[i] ^ key_bytes[j]);
      }
      return new String(res_bytes, StandardCharsets.UTF_8);
   }
}
```

VowelsStrategy.java:

```
class VowelsStrategy implements EncryptStrategy {
   private static Logger logger = Logger.getLogger(VowelsStrategy.class);
   final String englishVowels = "aeiouAEIOU";
   final String russianVowels = "аеёиоуыэюяАЕЁИОУЫЭЮЯ";
   String decryptionArray = *массив на 500+ слов*;
   String toEncrypt, toDecrypt;
   final static private String INITIALIZE = "";
   final static private String SPACE = " ";
   final static private int NOT_FOUND = -1;

   /**
    * Creating new object from string to encrypt
    * @param toEncrypt
    * <br/>
    * Use {@link VowelsStrategy#encrypt()} for encryption.
    * <br/>
    * And {@link VowelsStrategy#decrypt()} for decryption.
    */
   VowelsStrategy(String toEncrypt){
      this.toEncrypt = toEncrypt;
   }

   /**
    * Encrypting kdn.lab6.task1.encryption.VowelsStrategy object
    * @return encryptedString
    */
   @Override
   public String encrypt() {
      StringBuilder sb = new StringBuilder();
      for (char ch : toEncrypt.toCharArray()) {
         if (!isVowel(ch)) {
            sb.append(ch);
         }
      }
      toEncrypt = sb.toString();
      logger.info("File was encrypted with Vowels Deleting method");
      return toEncrypt;
   }

   /**
    * Decrypting kdn.lab6.task1.encryption.VowelsStrategy object
    * @return decryptedString
    */
   @Override
   public String decrypt() {
      boolean adden;
      toDecrypt = INITIALIZE;
      String[] wordsArray = decryptionArray.split(SPACE);
      String[] textToDecrypt = toEncrypt.split(SPACE);
      for(String wordToDecrypt : textToDecrypt){
         adden = false;
         VowelsStrategy encrypted = new VowelsStrategy(wordToDecrypt);
         for(String word : wordsArray){
            VowelsStrategy encryptedWord = new VowelsStrategy(word);
            if(encryptedWord.encrypt().toString().equalsIgnoreCase(encrypted.encrypt().toString())){
               toDecrypt += word + SPACE;
               adden = true;
               break;
            }
         }
         if(!adden){
            toDecrypt += wordToDecrypt + SPACE;
         }
      }
      logger.info("File was decrypted with Vowels Deleting method");
      return toDecrypt;
   }
```

```
    /**
     * Searching char 'ch' in array of vowels
     * @param ch
     * @return
     */
    private boolean isVowel(char ch) {
        String vowels = englishVowels + russianVowels;
        return vowels.indexOf(ch) != NOT_FOUND;
    }

}
```

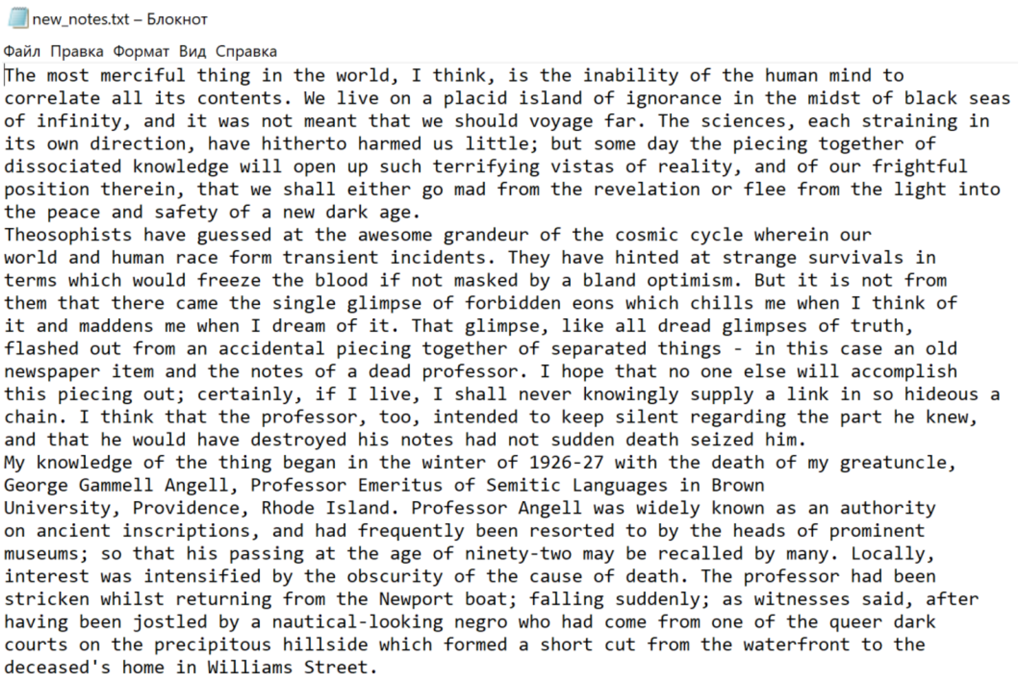
AtbashStrategy.java:

```
class AtbashStrategy implements EncryptStrategy {
    private static Logger logger = Logger.getLogger(AtbashStrategy.class);
    String toEncrypt, toDecrypt;
    int bias;

    /**
     *
     * @param toEncrypt
     * @param bias
     */
    AtbashStrategy(String toEncrypt, int bias){
        this.toEncrypt = toEncrypt;
        this.bias = bias;
    }
    @Override
    public String encrypt() {
        toEncrypt = toEncrypt.chars()
                .mapToObj(c -> (int)c)
                .map(c -> bias(c, bias, true))
                .collect(Collectors.joining());
        logger.info("File was encrypted with Atbash method");
        return toEncrypt;
    }
    @Override
    public String decrypt() {
        if(toDecrypt == null){
            toDecrypt = toEncrypt.chars()
                    .mapToObj(c -> (int)c)
                    .map(c -> bias(c, bias, false))
                    .collect(Collectors.joining());
        }else {
            toDecrypt = toDecrypt.chars()
                    .mapToObj(c -> (int)c)
                    .map(c -> bias(c, bias, false))
                    .collect(Collectors.joining());
        }
        logger.info("File was decrypted with Atbash method");
        return toDecrypt;
    }
    public String bias(int c, int bias, boolean add) {
        if (add) {
            c = (c + bias) % (Character.MAX_VALUE + 1);
        } else {
            c = (c - bias + Character.MAX_VALUE + 1) % (Character.MAX_VALUE + 1);
        }
        return String.valueOf((char) c);
    }
}
```

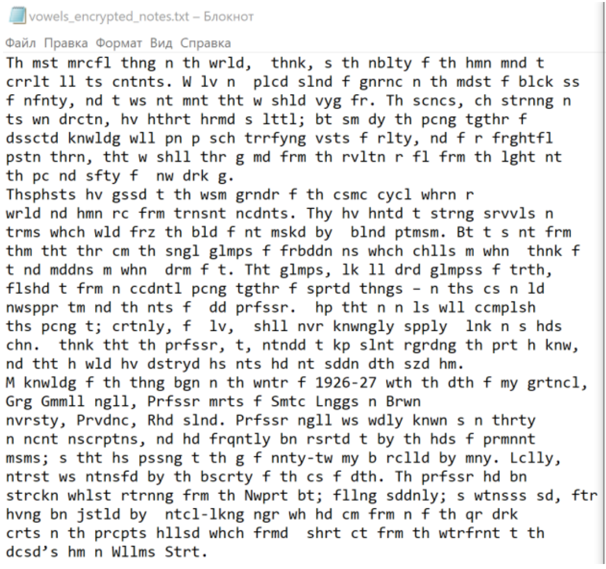
Main.java:

```
import kdn.lab6.task1.encryption.AtbashEncryption;
import kdn.lab6.task1.encryption.Encryption;
import kdn.lab6.task1.encryption.VowelsDelete;
import kdn.lab6.task1.encryption.XorEncryption;

public class Main {
  public static void main(String[] args) {
    final String initialPath = "new_notes.txt";
    final String key = ")%631yegd758YGUO+@*\"";
    final int bias = 5;

    String vowelsEncryptedPath = "vowels_encrypted_notes.txt";
    String vowelsDecryptedPath = "vowels_decrypted_notes.txt";
    String xorEncryptedPath = "xor_encrypted_notes.txt";
    String xorDecryptedPath = "xor_decrypted_notes.txt";
    String atbashEncryptedPath = "atbash_encrypted_notes.txt";
    String atbashDecryptedPath = "atbash_decrypted_notes.txt";


    VowelsDelete vowelsDelete = new VowelsDelete(initialPath);
    XorEncryption xorEncryption = new XorEncryption(initialPath, key);
    AtbashEncryption atbashEncryption = new AtbashEncryption(initialPath, bias);

    vowelsDelete.encrypt(vowelsEncryptedPath);
    vowelsDelete.decrypt(vowelsDecryptedPath);

    xorEncryption.encrypt(xorEncryptedPath);
    xorEncryption.decrypt(xorDecryptedPath);

    atbashEncryption.encrypt(atbashEncryptedPath);
    atbashEncryption.decrypt(atbashDecryptedPath);
  }
}
```

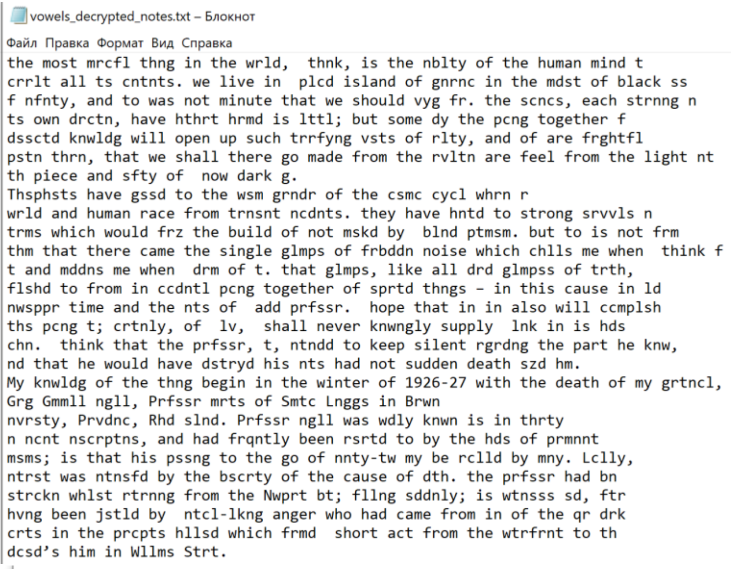
Рисунки с результатами работы программы



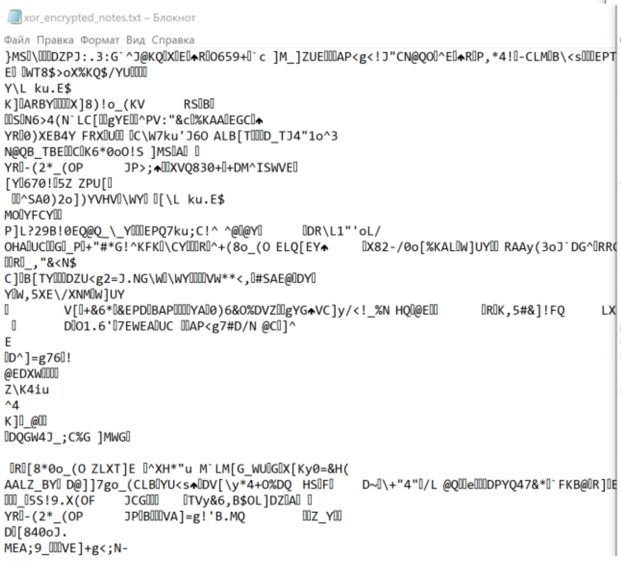
```
The most merciful thing in the world, I think, is the inability of the human mind to
correlate all its contents. We live on a placid island of ignorance in the midst of black seas
of infinity, and it was not meant that we should voyage far. The sciences, each straining in
its own direction, have hitherto harmed us little; but some day the piecing together of
dissociated knowledge will open up such terrifying vistas of reality, and of our frightful
position therein, that we shall either go mad from the revelation or flee from the light into
the peace and safety of a new dark age.
Theosophists have guessed at the awesome grandeur of the cosmic cycle wherein our
world and human race form transient incidents. They have hinted at strange survivals in
terms which would freeze the blood if not masked by a bland optimism. But it is not from
them that there came the single glimpse of forbidden eons which chills me when I think of
it and maddens me when I dream of it. That glimpse, like all dread glimpses of truth,
flashed out from an accidental piecing together of separated things - in this case an old
newspaper item and the notes of a dead professor. I hope that no one else will accomplish
this piecing out; certainly, if I live, I shall never knowingly supply a link in so hideous a
chain. I think that the professor, too, intended to keep silent regarding the part he knew,
and that he would have destroyed his notes had not sudden death seized him.
My knowledge of the thing began in the winter of 1926-27 with the death of my greatuncle,
George Gammell Angell, Professor Emeritus of Semitic Languages in Brown
University, Providence, Rhode Island. Professor Angell was widely known as an authority
on ancient inscriptions, and had frequently been resorted to by the heads of prominent
museums; so that his passing at the age of ninety-two may be recalled by many. Locally,
interest was intensified by the obscurity of the cause of death. The professor had been
stricken whilst returning from the Newport boat; falling suddenly; as witnesses said, after
having been jostled by a nautical-looking negro who had come from one of the queer dark
courts on the precipitous hillside which formed a short cut from the waterfront to the
deceased's home in Williams Street.
```


vowels_encrypted_notes.txt – Блокнот
Файл Правка Формат Вид Справка

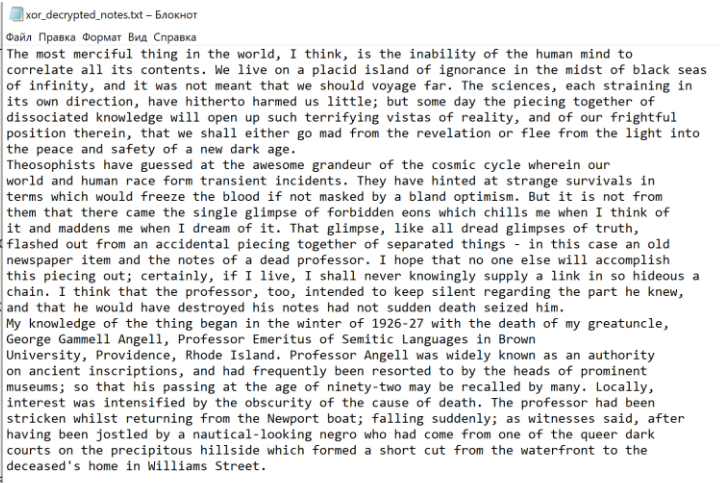
```
Th mst mrcfl thng n th wrld,  thnk, s th nblty f th hmn mnd t
crrlt ll ts cntnts. W lv n  plcd slnd f gnrnc n th mdst f blck ss
f nfnty, nd t ws nt mnt tht w shld vyg fr. Th scncs, ch strnng n
ts wn drctn, hv hthrt hrmd s lttl; bt sm dy th pcng tgthr f
dssctd knwldg wll pn p sch trrfyng vsts f rlty, nd f r frghtfl
pstn thrn, tht w shll thr g md frm th rvltn r fl frm th lght nt
th pc nd sfty f  nw drk g.
Thsphsts hv gssd t th wsm grndr f th csmc cycl whrn r
wrld nd hmn rc frm trnsnt ncdnts. Thy hv hntd t strng srvvls n
trms whch wld frz th bld f nt mskd by  blnd ptmsm. Bt t s nt frm
thm tht thr cm th sngl glmps f frbddn ns whch chlls m whn  thnk f
t nd mddns m whn  drm f t. Tht glmps, lk ll drd glmpss f trth,
flshd t frm n ccdntl pcng tgthr f sprtd thngs - n ths cs n ld
nwsppr tm nd th nts f  dd prfssr.  hp tht n n ls wll ccmplsh
ths pcng t; crtnly, f  lv,  shll nvr knwngly spply  lnk n s hds
chn.  thnk tht th prfssr, t, ntndd t kp slnt rgrdng th prt h knw,
nd tht h wld hv dstryd hs nts hd nt sddn dth szd hm.
M knwldg f th thng bgn n th wntr f 1926-27 wth th dth f my grtncl,
Grg Gmmll ngll, Prfssr mrts f Smtc Lnggs n Brwn
nvrsty, Prvdnc, Rhd slnd. Prfssr ngll ws wdly knwn s n thrty
n ncnt nscrptns, nd hd frqntly bn rsrtd t by th hds f prmnnt
msms; s tht hs pssng t th g f nnty-tw my b rclld by mny. Lclly,
ntrst ws ntnsfd by th bscrty f th cs f dth. Th prfssr hd bn
strckn whlst rtrnng frm th Nwprt bt; fllng sddnly; s wtnsss sd, ftr
hvng bn jstld by  ntcl-lkng ngr wh hd cm frm n f th qr drk
crts n th prcpts hllsd whch frmd  shrt ct frm th wtrfrnt t th
dcsd’s hm n Wllms Strt.
```


vowels_decrypted_notes.txt – Блокнот
Файл Правка Формат Вид Справка

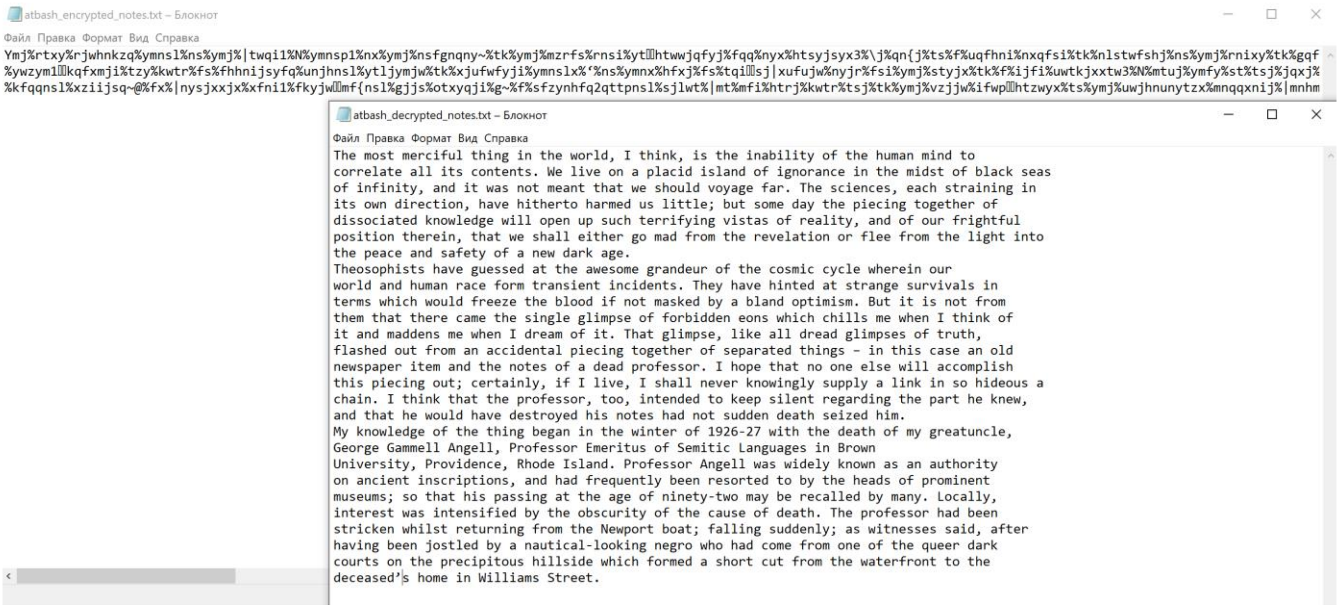
```
the most mrcfl thng in the wrld,  thnk, is the nblty of the human mind t
crrlt all ts cntnts. we live in  plcd island of gnrnc in the mdst of black ss
f nfnty, and to was not minute that we should vyg fr. the scncs, each strnng n
ts own drctn, have hthrt hrmd is lttl; but some dy the pcng together f
dssctd knwldg will open up such trrfyng vsts of rlty, and of are frghtfl
pstn thrn, that we shall there go made from the rvltn are feel from the light nt
th piece and sfty of  now dark g.
Thsphsts have gssd to the wsm grndr of the csmc cycl whrn r
wrld and human race from trnsnt ncdnts. they have hntd to strong srvvls n
trms which would frz the build of not mskd by  blnd ptmsm. but to is not frm
thm that there came the single glmps of frbddn noise which chlls me when  think f
t and mddns me when  drm of t. that glmps, like all drd glmpss of trth,
flshd to from in ccdntl pcng together of sprtd thngs - in this cause in ld
nwsppr time and the nts of  add prfssr.  hope that in in also will ccmplsh
ths pcng t; crtnly, of  lv,  shall never knwngly supply  lnk in is hds
chn.  think that the prfssr, t, ntndd to keep silent rgrdng the part he knw,
nd that he would have dstryd his nts had not sudden death szd hm.
My knwldg of the thng begin in the winter of 1926-27 with the death of my grtncl,
Grg Gmmll ngll, Prfssr mrts of Smtc Lnggs in Brwn
nvrsty, Prvdnc, Rhd slnd. Prfssr ngll was wdly knwn is in thrty
n ncnt nscrptns, and had frqntly been rsrtd to by the hds of prmnnt
msms; is that his pssng to the go of nnty-tw my be rclld by mny. Lclly,
ntrst was ntnsfd by the bscrty of the cause of dth. the prfssr had bn
strckn whlst rtrnng from the Nwprt bt; fllng sddnly; is wtnsss sd, ftr
hvng been jstld by  ntcl-lkng anger who had came from in of the qr drk
crts in the prcpts hllsd which frmd  short act from the wtrfrnt to th
dcsd’s him in Wllms Strt.
```


xor_encrypted_notes.txt – Блокнот
Файл Правка Формат Вид Справка

```
}MS□\□□□DZPJ:.3:G`^J@KQ□X□E□♠R□O659+□`c ]M_]ZUE□□□AP<g<!J"CN@QO□^E□♠R□P,*4!□-CLM□B\<s□□□EPT
E□ □WT8$>oX%KQ$/YU□□□□
Y\L ku.E$
K]□ARBY□□□□X]8)!o_(KV      RS□B□
□□S□N6>4(N`LC[□□gYE□□^PV:"&c□%KAA□EGC□♠
YR□0)XEB4Y FRX□U□□ □C\W7ku'J6O ALB[T□□□D_TJ4"1o^3
N@QB_TBE□□C□K6*0oO!S ]MS□A□ □
YR□-(2*_(OP      JP>;♠□□XVQ830+□+DM^ISWVE□
[Y□670!□5Z ZPU[□
 □□^SA0)2o])YVHV□\WY□ □[\L ku.E$
MO□YFCY□□
P]L?29B!0EQ@Q_\_Y□□□EPQ7ku;C!^ ^@□@Y□      □DR\L1"'oL/
OHA□UC□□G□_P□+"#*G!^KFK□\CY□□□R□^+(8o_(O ELQ[EY♠      □X82-/0o[%KAL□W]UY□□ RAAy(3oJ`DG^□RR(
□□R□_,"&<N$
C]□B[TY□□□DZU<g2=J.NG\W□\WY□□□□VW**<,□#SAE@□DY□
Y□W,5XE\/XNM□W]UY
□        V[□+&6*□&EPD□BAP□□□□YA□0)6&O%DVZ□□gYG♠VC]y/<!_%N HQ□@E□□      □R□K,5#&]!FQ     LX
 □       D□O1.6'□7EWEA□UC □□AP<g7#D/N @C□]^
E
□D^]=g76□!
@EDXW□□□□
Z\K4iu
^4
K]□_@□□
□DQGW4J_;C%G ]MWG□

 □R□[8*0o_(O ZLXT]E □^XH*"u M`LM[G_WU□G□X[Ky0=&H(
AALZ_BY□ D@]]7go_(CLB□YU<s♠□DV[\y*4+O%DQ  HS□F□     D~□\+"4"□/L @Q□□e□□□DPYQ47&*□`FKB@□R]□E
□□□_□5S!9.X(OF     JCG□□□     □TVy&6,B$OL]DZ□A□ □
YR□-(2*_(OP      JP□B□□□VA]=g!'B.MQ         □□Z_Y□□
D□[840oJ.
MEA;9_□□□VE]+g<;N-
```


xor_decrypted_notes.txt – Блокнот
Файл Правка Формат Вид Справка

```
The most merciful thing in the world, I think, is the inability of the human mind to
correlate all its contents. We live on a placid island of ignorance in the midst of black seas
of infinity, and it was not meant that we should voyage far. The sciences, each straining in
its own direction, have hitherto harmed us little; but some day the piecing together of
dissociated knowledge will open up such terrifying vistas of reality, and of our frightful
position therein, that we shall either go mad from the revelation or flee from the light into
the peace and safety of a new dark age.
Theosophists have guessed at the awesome grandeur of the cosmic cycle wherein our
world and human race form transient incidents. They have hinted at strange survivals in
terms which would freeze the blood if not masked by a bland optimism. But it is not from
them that there came the single glimpse of forbidden eons which chills me when I think of
it and maddens me when I dream of it. That glimpse, like all dread glimpses of truth,
flashed out from an accidental piecing together of separated things - in this case an old
newspaper item and the notes of a dead professor. I hope that no one else will accomplish
this piecing out; certainly, if I live, I shall never knowingly supply a link in so hideous a
chain. I think that the professor, too, intended to keep silent regarding the part he knew,
and that he would have destroyed his notes had not sudden death seized him.
My knowledge of the thing began in the winter of 1926-27 with the death of my greatuncle,
George Gammell Angell, Professor Emeritus of Semitic Languages in Brown
University, Providence, Rhode Island. Professor Angell was widely known as an authority
on ancient inscriptions, and had frequently been resorted to by the heads of prominent
museums; so that his passing at the age of ninety-two may be recalled by many. Locally,
interest was intensified by the obscurity of the cause of death. The professor had been
stricken whilst returning from the Newport boat; falling suddenly; as witnesses said, after
having been jostled by a nautical-looking negro who had come from one of the queer dark
courts on the precipitous hillside which formed a short cut from the waterfront to the
deceased's home in Williams Street.
```


atbash_encrypted_notes.txt – Блокнот
Файл Правка Формат Вид Справка

```
Ymj%rtxy%rjwhnkzq%ymnsl%ns%ymj%|twqi1%N%ymnsp1%nx%ymj%nsfgnqny~%tk%ymj%mzrfs%rnsi%yt□□htwwjqfyj%fqq%nyx%htsyjsyx3%\j%qn{j%ts%f%uqfhni%nxqfsi%tk%nlstwfshj%ns%ymj%rnixy%tk%gqf
%ywzym1□□kqfxmji%tzy%kwtr%fs%fhhnijsyfq%unjhnsl%ytljymjw%tk%xjufwfyji%ymnslx%‘%ns%ymnx%hfxj%fs%tqi□□sj|xufujw%nyjr%fsi%ymj%styjx%tk%f%ijfi%uwtkjxxtw3%N%mtuj%ymfy%st%tsj%jqxj%
%kfqqnsl%xziijsq~@%fx%|nysjxxjx%xfni1%fkyjw□□mf{nsl%gjjs%otxyqji%g~%f%sfzynhfq2qttpnsl%sjlwt%|mt%mfi%htrj%kwtr%tsj%tk%ymj%vzjjw%ifwp□□htzwyx%ts%ymj%uwjhnunytzx%mnqqxnij%|mnhm
```

atbash_decrypted_notes.txt – Блокнот
Файл Правка Формат Вид Справка

```
The most merciful thing in the world, I think, is the inability of the human mind to
correlate all its contents. We live on a placid island of ignorance in the midst of black seas
of infinity, and it was not meant that we should voyage far. The sciences, each straining in
its own direction, have hitherto harmed us little; but some day the piecing together of
dissociated knowledge will open up such terrifying vistas of reality, and of our frightful
position therein, that we shall either go mad from the revelation or flee from the light into
the peace and safety of a new dark age.
Theosophists have guessed at the awesome grandeur of the cosmic cycle wherein our
world and human race form transient incidents. They have hinted at strange survivals in
terms which would freeze the blood if not masked by a bland optimism. But it is not from
them that there came the single glimpse of forbidden eons which chills me when I think of
it and maddens me when I dream of it. That glimpse, like all dread glimpses of truth,
flashed out from an accidental piecing together of separated things - in this case an old
newspaper item and the notes of a dead professor. I hope that no one else will accomplish
this piecing out; certainly, if I live, I shall never knowingly supply a link in so hideous a
chain. I think that the professor, too, intended to keep silent regarding the part he knew,
and that he would have destroyed his notes had not sudden death seized him.
My knowledge of the thing began in the winter of 1926-27 with the death of my greatuncle,
George Gammell Angell, Professor Emeritus of Semitic Languages in Brown
University, Providence, Rhode Island. Professor Angell was widely known as an authority
on ancient inscriptions, and had frequently been resorted to by the heads of prominent
museums; so that his passing at the age of ninety-two may be recalled by many. Locally,
interest was intensified by the obscurity of the cause of death. The professor had been
stricken whilst returning from the Newport boat; falling suddenly; as witnesses said, after
having been jostled by a nautical-looking negro who had come from one of the queer dark
courts on the precipitous hillside which formed a short cut from the waterfront to the
deceased’s home in Williams Street.
```

Вывод: приобрел навыки применения паттернов проектирования при решении практических задач с использованием языка Java.