

Home Work #5 ~ Chat Program Design

Shahar Perets

29.12.2024

Screen.java

```
1 public class Screen {
2     private Screen();
3     public Screen getScreen(); // singleton
4     public void drawMessage(List<Message> messages); // calls (foreach) message.draw()
5 }
6 public void drawUI();
```

package messages

messages.Message.java

```
1 public class Message extends IDrawable {
2     protected User user;
3     protected TimeStamp sendingTime;
4
5     public Message(User user, TimeStamp sendingTime);
6     public void draw(); // calls getUser(), getSendingTime().format()
7     private User getUser();
8     private TimeStamp getSendingTime();
9     public deleteMessage();
10
11     abstract boolean[] encode();
12     abstract static<T extends Message> decode(boolean[] dataStream);
13 }
```

messages.PlyableMessage.java

```
1 public abstract class PlyableMessage extends Message {
2     private void startPlaying(); // would be binded into a button created in draw()
3     private void stopPlaying(); // same here
4 }
```

Note: all of the constructors below that in the package calls the parent constructor.

messages.TextMessage.java

```
1 public class TextMessage extends Message {
2     private String text;
3     public TextMessages(String sender, TimeStamp sendingTime, String recipient, String text);
4     @Override
5     public draw() {
6         super.draw();
7         // additional implementations for drawing a text message
8     }
9     @Override getContenWidth(), getContenHeight(), decode(), encode(), download()
10 }
```

messages.ImageMessage.java

```
1 public class ImageMessage extends Message {
2     private RGBA[][][] image
3     public TextMessages(User user, TimeStamp sendingTime, RGBA[][] image);
4     @Override
5     public draw() {
```

```

6         super.draw();
7         // additional implementations for drawing an image
8     }
9     @Override getContentView(), getContentViewHeight(), decode(), encode(), download()
10 }

```

messages.VocalMessage.java

```

1 public class ImageMessage extends Message {
2     private double[][] discreteFrequencyOverTimeMiliSec;
3     public TextMessages(
4         User user,
5         Timestamp sendingTime,
6         double[][] discreteFrequencyOverTimeMiliSec
7     );
8     @Override
9     public draw() {
10         super.draw();
11         // additional implementations for drawing a vocal message
12     }
13     private double calculateDurationSeconds();
14
15     private static String createAnAISummary(double[][] discreteFrequencyOverTimeMiliSec); // because we live in 2025
16
17     @Override startPlaying(), stopPlaying(), getContentView(), getContentViewHeight(), decode(), encode(), download()
18 }

```

messages.VideoMessage.java

```

1 public class VideoMessage extends PlyableMessage {
2     private RGBA[][][] imagesOverTime;
3     public TextMessages(
4         User user,
5         Timestamp sendingTime,
6         String recipient,
7         RGBA[][][] imagesOverTime
8     );
9     @Override
10    public draw() {
11        super.draw();
12        // additional implementations for drawing a vocal message
13    }
14    @Override startPlaying(), stopPlaying(), getContentView(), getContentViewHeight(), decode(), encode(), download()
15 }

```

messages.DocumentMessage.java

```

1 public class VideoMessage extends Message {
2     private boolean[] fileBytes;
3     public TextMessages(
4         User user,
5         Timestamp sendingTime,
6         String recipient,
7         boolean[] fileBytes
8     );
9     @Override
10    public draw() {
11        super.draw();
12        // additional implementations for drawing a vocal message
13    }
14    @Override getContentView(), getContentViewHeight(), decode(), encode(), download()
15 }

```

package utils

utils.RGBA.java

```

1 public record RGBA(double red, double green, double blue, double alpha);

```

utils.Timestamp.java

```

1 public record Timestamp(int UNIXTime) {
2     public String format();
3     public int getUNIXTime();
4 }

```

package utils.decoder

utils.decoder.IDecoder.java

```

1 public interface IDecoder {
2     boolean[] decode(boolean[] data);
3     boolean[] encode(boolean[] data);
4 }

```

utils.decoder.AESDecoder.java

```

1 public class AESDecoder implements IDecoder {
2     @Override encode, decode
3 }

```

utils.decoder.AESDecoder.java

```

1 public class DESDecoder implements IDecoder {
2     @Override encode, decode
3 }

```

AdditionalNotes.md

- כאשר ייכתב @Override A, B... הכוונה היא שנדרוס ונממש את הפונקציות באותו המקום.

- הקוד ניתן להרחבה, כי בעת הצורך ליצירת סוג הודעה חדש, כי ניתן להרחיב את המחלקה האבסטרקטית Message, ללא שימוש רפטטיבי בקוד שכבר ממומש בה, וכל החלקים האחרים בקוד יוכלו לתפקד באופן פולימורפי עם סוג ההודעה החדש כמו שאר הסוגים. נוסף על כך מימוש ההצפנה יוכל להיות פשוט במיוחד משום שהמתמטיקה תבוצע בפונקציה כתובה, בעוד רק צריך לעטוף את אותה הפונקציה בלוגיקה פר־הודעה שקשורה בהצפנה.

- יידרש אך ורק לממש את המתודות האבסטרקטיות של Message (אם כי הוספת פונקציות נוספות למטרות פיצול קוד ומטרה יחידה מוגדרת היטב ל־draw() כמובן יועדפו).