

תרגיל בית מספר 4

שם הקורס: עיצוב ותכנות מונחי עצמים להנדסה

מספר קורס: 157109

נושא: עקרונות עיצוב SOLID

שאלה 1

נתונות הדרישות הבאות למערכת העוסקת בהצפנה ובפענוח:

המערכת מכילה מקודדים (Encoder) ומפענחים (Decoder). מקודדים יודעים לבצע פעולת encode ומפענחים יודעים לבצע פעולת decode.

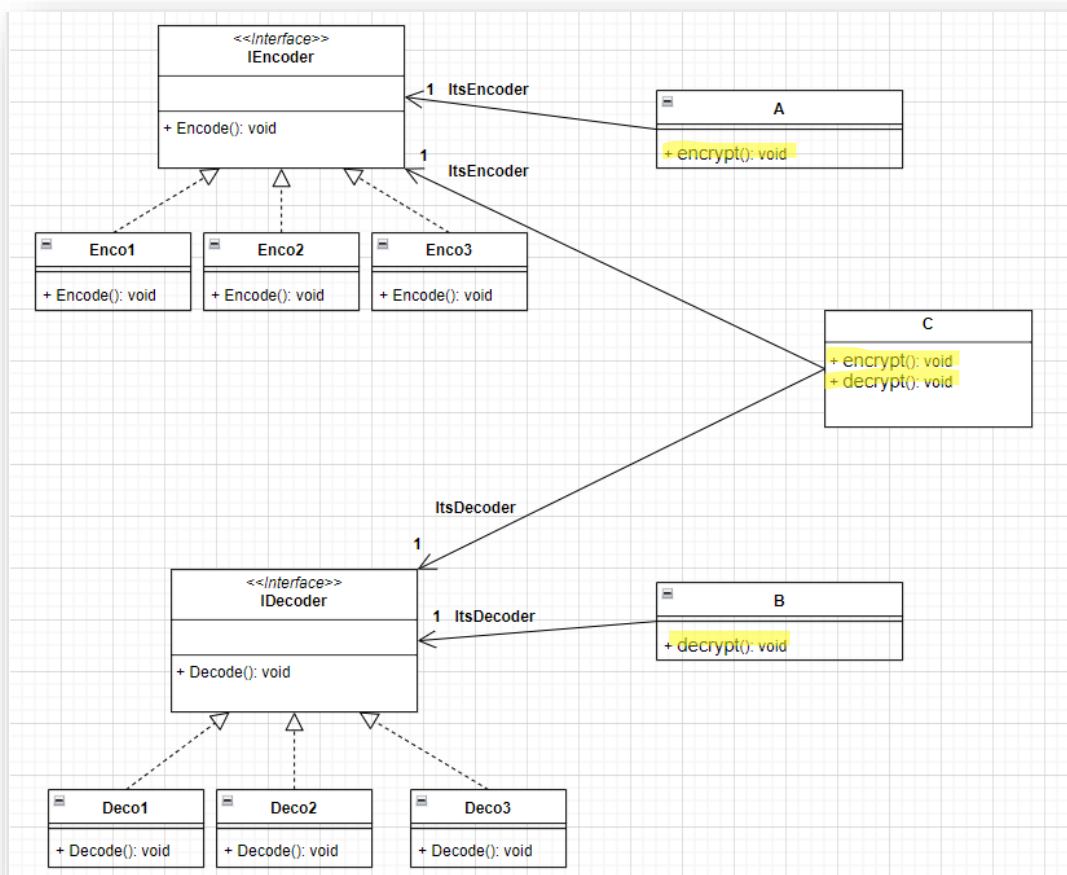
ישנם שלשה טיפוסים שדרושים להם אובייקטים מסוג Encoder ו/או Decoder. נכנה אותם בשמות A, B, C. עוסק בהצפנה בלבד. הוא יודע לבצע פעולת encrypt ולשם כך דרושים לו אובייקטים מסוג Encoder, והוא משתמש בפעולת encode שלהם.

B עוסק בפענוח בלבד. הוא יודע לבצע פעולת decrypt ולשם כך דרושים לו אובייקטים מסוג Decoder, והוא משתמש בפעולת decode שלהם.

C עוסק גם בהצפנה וגם בפענוח. הוא יודע לבצע פעולת encrypt וגם פעולת decrypt ולשם כך דרושים לו אובייקטים מסוג Encoder וגם אובייקטים מסוג Decoder, והוא משתמש בהתאמה בפעולות encode, decode שלהם.

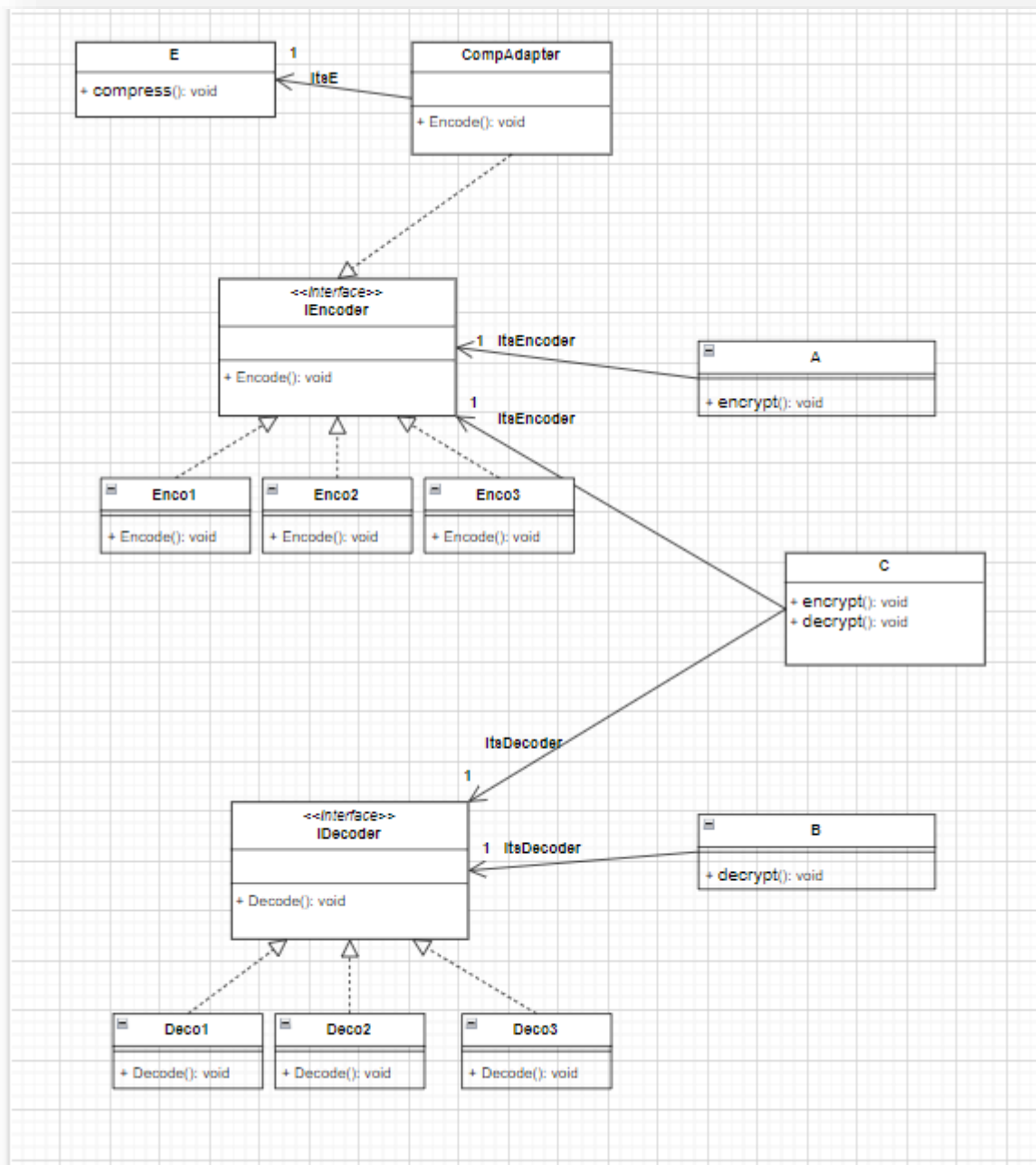
ישנם שלשה סוגי מקודדים: Enco1, Enco2, Enco3 ובהתאמה שלושה סוגי מפענחים: Deco1, Deco2, Deco3.

סעיף א: צייר תרשים מחלקות העומד בדרישות ומקיים את עקרונות SOLID



סעיף ב: המתכנתים מצאו ברשת חבילת תכנה המכילה מחלקה מקומפלת E, שיש לה את הפעולה compress (במקום encode). אנו רוצים לשלב את E כמקודד נוסף, אך אין באפשרותנו לנגוע בקוד שלו, או אפילו לקמפל אותו מחדש. הוסף את E למערכת (צייר תרשים חדש) כך שתעמוד בדרישות ותקיים את עקרונות SOLID. כתוב באיזו תבנית עיצוב השתמשת כדי לעשות זאת (ציין את שם התבנית). (הערת הבנה: אם מוסיפים למערכת מקודד חדש, אזי יש להוסיף גם את המפענח המתאים לו, אבל בסעיף זה אינך נדרש להוסיף גם את המפענח, אלא רק את המקודד).

השתמשו ב-object Adapter:



סעיף ג: (התעלם מסעיף ב): מכיוון שכל Encoder ספציפי עובד לפי אלגוריתם הצפנה מסוים, וה- decoder המתאים לו חייב לעבוד לפי אותו אלגוריתם וכ"תמונת מראה" של תהליך ההצפנה שהתבצע לפי אותו אלגוריתם, הגענו למסקנה שההפרדה של פעולות encode ו decode לשתי מחלקות שונות (Encoder, Decoder) אינה נכונה והיא מפירה אחד מעקרונות SOLID. עליך לציין איזה עיקרון, מעקרונות SOLID, מופר.

אין פה הפרה משמעותית של עקרונות SOLID, בעיקר שימוש לא נכון בעקרונות SRP ו ISP, עקרונות האחריות היחידה והפרדת הממשקים.

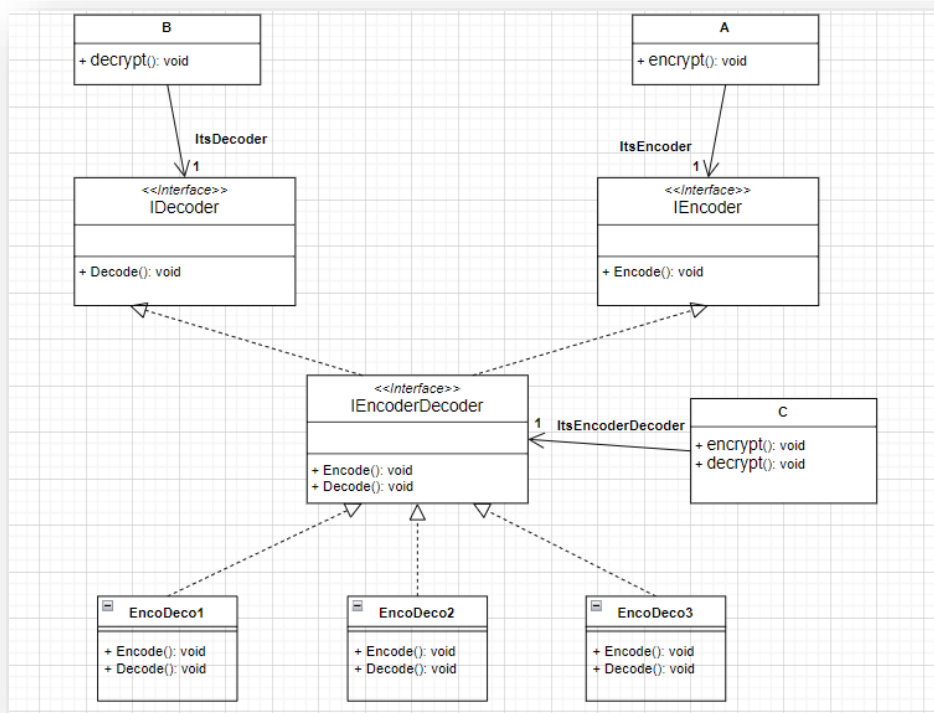
העקרון שנראה שמתממש פה בצורה לא נכונה ומופר הוא עקרון הפרדת הממשקים (ISP, Interface Segregation Principle) מכיוון שעל פי העקרון הנ"ל אנחנו צריכים לחלק את הקוד בצורה כזו שבמקום לבנות ממשק גדול אחד שמכיל מגוון פונקציות, עדיף לחלק אותו למספר ממשקים קטנים וממוקדים, וכך מחלקות יוכלו לממש את הפונקציונליות שהם זקוקים לה, אך פה המימוש לא שלם -מכיוון שבמקרה שלנו, כאשר Encoder ו-Decoder מופרדים לשתי מחלקות שונות, ישנה בעיה עבור מחלקות שצריכות לבצע גם הצפנה וגם פענוח. במקום שיהיה ממשק אחד שמכיל גם את פעולת encode וגם את פעולת decode, יש הפרדה כפויה בין השניים.

מכיוון שכל אלגוריתם הצפנה דורש את אותו אלגוריתם לפענוח, נכון יותר לייצר ממשק אחד שמכיל את שתי הפעולות. כך אנו ממזערים את מספר הממשקים ומוודאים שהלקוח לא צריך להשתמש במספר ממשקים שונים עבור פעולה שנעשית ע"י אותו אלגוריתם.

בנוסף עקרון שנראה בעייתי פה הוא עקרון האחריות היחידה, שאומר שכל מחלקה צריכה להיות אחראית על תפקיד אחד בלבד, אבל במקרה הזה המחלקות לא שלמות ולא מממשות בצורה מלאה את התפקיד שלהן, דבר שמפר את האחריות היחידה שלהן ותולה אותן אחת בשנייה.

לכן, הוחלט לאחד כל זוג מחלקות EncoderX, DecoderX למחלקה אחת שתיקרא EncoderDecoder.X (לדוגמה: במקום סוג מקודד ספציפי Enco1 ומפענח ספציפי Deco1 מתאים לו Deco1 (ראה לעיל) כשתי מחלקות נפרדות, תהיה לנו רק מחלקה אחת שתיקרא EncoDeco1. מחלקה זו יודעת לבצע פעולות encode וגם פעולות decode).

עדכן את המערכת (צייר תרשים חדש) כך שתעמוד בדרישות ותקיים את עקרונות SOLID. ציין באיזה עיקרון/ות היית צריך להתחשב. בעדכון המערכת התחשבנו בשתי עקרונות SRP & ISP, מכיוון שהיינו צריכים לממש את הפעולות בצורה של הפרדת הממשקים ל encoder ו initialization שלהם על ידי ממשק נוסף, והשתמשנו בעיקרון ה SRP כאשר איחדנו את המערכת וייצרנו אחריות יחידה שמממשת את התפקיד של מפענח וזוהי האחריות היחידה והשלמה שלה.



שאלה 2

נתונות הדרישות הבאות למערכת לעיבוד קול:

ישנה קבוצה של יונקים (Mammal). כל Mammal מכיל פונקציה Speak שמייצרת ייצוג של קול של החיה (מסוג RawData).

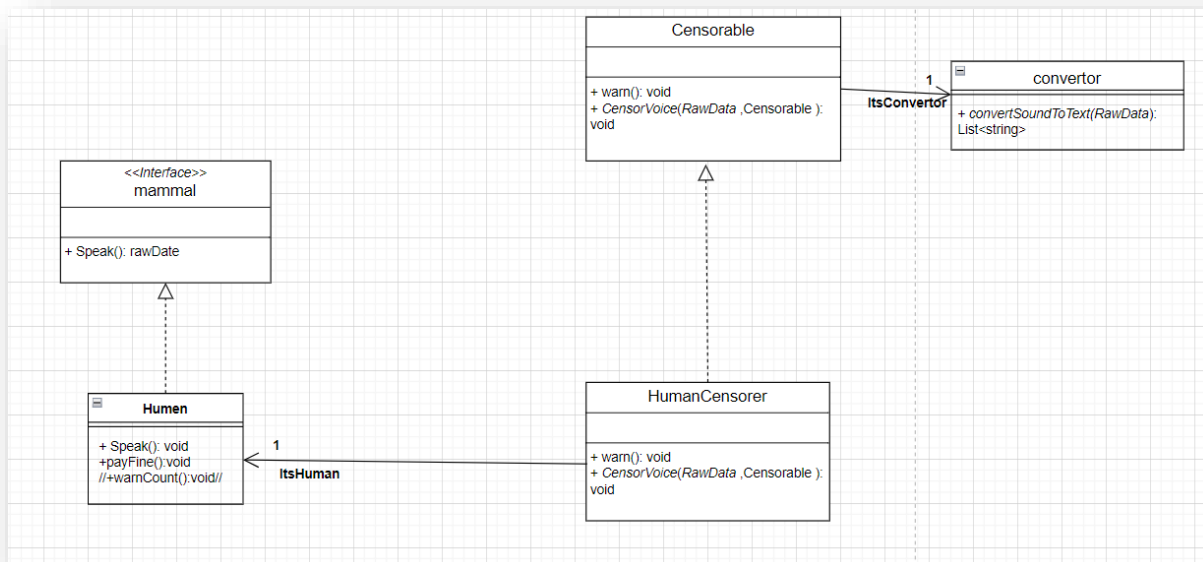
בן אדם (Human) הוא סוג של Mammal. כל הפעלה של פונקציה Speak על ידי Human תלווה בתהליך לניתוח קול. אם הקול מכיל מילים מסויימות (בהפעלות שונות ייתכנו מילים שונות) ה Human יקבל אזהרה. בן אדם שקיבל 3 אזהרות יפעיל פונקציה payFine.

מבין היונקים, ניתן להזהיר רק בני אדם. ייתכן שיש מחלקות נוספות, שאינן יונקים, שניתן להזהיר אותן (לדוגמא תוכי, פטיפון).

על מנת לעמוד בדרישות מוגדרות הפונקציות הבאות (תצטרכו להחליט באיזו מחלקה לשים כל אחת מהן):

- `List<String> convertSoundToText (RawData voice)`: returns a list of words that the voice contains
- `void CensorVoice(RawData voice, Censorable client)`: Analyzes the voice and if it contains certain words, *warns* the client (You should decide what should be the client's type)
- `void warn()`: Updates the object that it has been *warned*.

צייר תרשים מחלקות העומד בדרישות ומקיים (לא מפר) את עקרונות SOLID. שים לב שעליך לשבץ את הפונקציות שתוארו, למחלקות בתרשים (בהתאם לעקרון SRP).



המערכת בתרשים מממשת את הדרישות ומיישמת בצורה נכונה את עקרונות SOLID

הקשר בין HumanCensorer לHUMAN הוא קשר שך directed association , בצורה כזו ש human הוא בעצם רשימת בני האדם שניתן להזהיר אותם , במחלקה humanCensorer בדומה לדוגמה משיעור 1 , בו דיברנו על רשימת הסרטים שאדם ראה ועל הקשר בין מחלקה A לב .

HUMAN הוא סוג של יונק ולכן – הקשר ביניהם מימוש הפונקציות בHUMAN נבחרו לפי התבנית של SRP שנתונה במצגת.

קונברטור נבחר לייצג מחלקה נפרדת כדי שנוכל לעשות בו שימוש חוזר (לדוגמא פטיפון). על מנת שנוכל להזהיר יותר ישויות מHUMAN הוצאנו את CENSORABLE . הקשר בניהם הוא ASSOCIATION (שימוש של קונברטר באזהרה).

HUMANCENSORER – מימוש ספציפי של האזהרה לHUMAN.