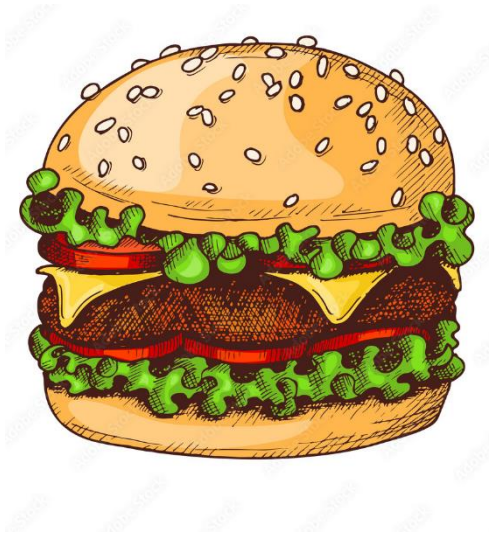
שאלה 1 – תכן בשפת C++



בשאלה זו נרצה ליצור קוד המדמה המבורגר (Hamburger).

המבורגר מורכב מלחמנייה עליונה (Upper), לחמנייה תחתונה (Lower), ומגוון רחב של תוספות (FoodItem).

לכל תוספת יש רמת טעם המיוצגת ע"י מספר שלם חיובי. התוספות הקיימות:

- קציצת בשר (Meat) במשקל של 160/220/260/320 גרם.
- ירקות (Vegetable):
  - עגבנייה (Tomato).
  - מלפפון רגיל (Cucumber) ומלפפון חמוץ (Pickle).
  - בצל (Onion).
  - תירס (Corn).
  - חסה (Lettuce).
- גבינות (Cheese):
  - צ'דר (Cheddar).
  - מוצרלה (Mozzarella).
  - אמריקן (American).
- בייקון (Bacon).

לכל המבורגר נגדיר את רמת הטעם שלו (Taste) ואת רמת האיכות שלו (Quality).

- רמת הטעם מוגדרת להיות סכום רמות הטעם של כל התוספות שנמצאות בהמבורגר.
- רמת האיכות מוגדרת להיות רמת הטעם חלקי כמות התוספות (שימו לב שזהו לא בהכרח מספר שלם).

הערות:

- הלחמניות לא נחשבות ברמת הטעם וברמת האיכות.
- הלחמנייה העליונה היא אופציונלית (ניתן להוסיף/להוריד אותה לפי הרצון ובכל עת) בעוד שהלחמנייה התחתונה היא הכרחית.
- לא ניתן להוריד/להוסיף לחמנייה עליונה פעמיים.
- לכל המבורגר חייבת להיות לפחות קציצת בשר אחת.
- כל סוג של תוספת (קציצה/עגבנייה/מלפפון וכו'...) בעל רמת טעם שהיא מספר שרירותי לבחירתכם.
- תוספות מאותו סוג תמיד בעלות אותה רמת טעם. כלומר כל העגבניות הן בעלות אותה רמת טעם, כל המלפפונים הם בעלי אותה רמת טעם, וכך הלאה.
- עקב סובייקטיביות הטעם האנושי, לא ייתכן שכל סוגי התוספות הם בעלי אותה רמת טעם (כלומר קיימים שני סוגי תוספות בעלי רמת טעם שונה).
- ייתכנו כפילויות בתוספות, לדוגמה, המבורגר יכול להכיל בייקון פעמיים.
- בשאלה זו נתעלם מרטבים.
- אם אתם טבעוניים/צמחוניים/אלרגיים הניחו כי הלחמניות, הקציצות, וכל התוספות עומדות בתנאים אלו (אם זה קשה לכם – תתמודדו).

לפניכם ממשק ומימוש חלקי של מחלקת FoodItem, והממשק של מחלקת Hamburger.

```
class FoodItem {
    int m_taste;
public:
    FoodItem(int taste) : m_taste(taste) {}

    int Taste() const {
        return m_taste;
    }

    virtual ~FoodItem() = default;
    virtual shared_ptr<FoodItem> clone()  const = 0;
};

class Hamburger {
public:
    Hamburger(const list<shared_ptr<FoodItem>>& contents, bool upper);

    void addContent(const shared_ptr<FoodItem>& item);

    int Taste() const;
    float Quality() const;

    bool addUpperBread();
    bool removeUpperBread();
};
```

א. ממשו את המחלקה Hamburger.

בנאי:

- מקבל רשימה של מצביעים חכמים לתוספות וערך בוליאני המציין האם לאתחל את ההמבורגר עם לחמנייה עליונה.
- אם אין קציצה (רגילה או טבעונית) בהמבורגר שנוצר הוא לא חוקי ויש לזרוק חריגת MeatlessBurger.

addContent:

- מקבלת מצביע חכם לתוספת ומוסיפה אותה לשאר התוספות שכבר קיימות.
- ההמבורגר צריך לשמור את התוספות שבו לפי הסדר שבו נוספו.

Taste:

- מחזירה את רמת הטעם של ההמבורגר.
- הקציצה הראשונה שהתווספה לא נחשבת ברמת הטעם של ההמבורגר.

Quality:

- מחזירה את רמת האיכות של ההמבורגר.
- הקציצה הראשונה שהתווספה כן נחשבת ברמת האיכות של ההמבורגר.

addUpperBread:

- מוסיפה את הלחמנייה העליונה, מחזירה ערך בוליאני המציין אם היא הצליחה או לא.
- לא ניתן להוסיף לחמנייה עליונה נוספת.

removeUpperBread:

- מורידה את הלחמנייה העליונה, מחזירה ערך בוליאני המציין אם היא הצליחה או לא.
- לא ניתן להוריד לחמנייה עליונה כשאין כזאת.

יש לממש כל פונקציה/מחלקת עזר שאתם מוסיפים, ואין צורך לממש את יתר המחלקות שב-UML.

אין להוסיף דבר לממשק של FoodItem ושל Hamburger. ההוראה תקפה גם לסעיפים הבאים. ניתן להוסיף מתודות פרטיות למחלקה Hamburger.

ב. שרטטו תכן העונה לצרכי כל המערכת בעזרת UML. ציינו את שמה של כל תבנית תכן שהשתמשתם בה.

ג. ממשו את הפונקציה החיצונית

```
void sortByQuality (vector<Hamburger>& hamburgers);
```

הפונקציה מקבלת וקטור של המבורגרים וממיינת אותו לפי רמת האיכות של ההמבורגרים.

- יש לממש כל פונקציה/מחלקת עזר שאתם מוסיפים.
- אם אתם צריכים לשנות משהו במחלקת Hamburger ציינו זאת.

ד. ממשו את קציצות הבשר ואת הגבינות. ממשו רק את מה שדרוש עבור מימוש התכן ולא מעבר לכך.

- ניתן להניח שמשקל הקציצות תקין, אין צורך לבדוק אותו.

כעת נרצה להוסיף צ'יפס (Fries) להמבורגר ולהגיש את שניהם על מגש (Tray).

- צ'יפס יכול להיות עשוי מזנים שונים של תפוחי אדמה (Potato): באטר (Batter), ראטה (Rata), ויולדי (Vivaldi) וכו'.
- מגש חייב להכיל המבורגר אחד ומנת צ'יפס אחת.

ה. שרטטו תכן העונה לצרכי כל המערכת בעזרת UML. ציינו את שמה של כל תבנית תכן שהשתמשתם בה. אין צורך לציין תבניות שהשתמשתם בהן בסעיף ב'.