# תרגיל חופשת סוכות

# בניית משחק בתכנות מונחה עצמים

## תיכון גוטמן – כיתה יא

## מורה : דפנה פילוסוף



OOP - תכנות מונחה עצמים הינו תכנות שמבוסס מחלקות ועצמים והוא אחד הנושאים המרכזיים בחומר הלימוד שלנו.

כדי לתרגל ולהעמיק את הידע בנושא , אתם תתבקשו לתכנן ולתכנת משחק קלפים המבוסס על עקרונותיו.

חשוב להדגיש:

• מומלץ להתמודד עם התרגיל ללא שימוש בכלי AI במטרה להבין את הנושא.

• תרגיל הזה הוא תרגיל עם ציון .

**תיאור המשחק:**

המשחק הוא משחק מחשב בין שני שחקנים (יתכן יותר משניים או של שחקן מול מחשב).

בתחילת המשחק, כל שחקן מקבל 7 קלפים )תכונה שאפשר לשנות).

בכל תור כל שחקן בוחר קלף ראשון מתוך הקלפים ביד שלו.

השחקן עם הקלף הגבוה זוכה בשני הקלפים ושם אותם בחבילת הזכיות - אם יש שוויון הקלפים חוזרים לחפיסה.

המשחק נגמר אחרי 7 סבבים )כשהשחקנים מסיימים את כל הקלפים שבידיהם(.

מנצח המשחק הוא השחקן אשר לו בחפיסת הזכיות סכום הקלפים שערכם הוא הגבוה ביותר .

**תיאור מהלך התוכנה של המשחק:**

המשחק יתחיל בהוד עה למשתמש " Welcome to my cards game!! ."

המשחק יבנה חפיסת קלפים, יצור שני שחקנים ויחלק לכל שחקן 7 קלפים (בצורה אקראית(.

לאחר סיום האתחול, המשחק יריץ לולאה המפעילה את תורות המשחק עד סיום המשחק.

בכל תור כל שחקן פותח קלף, המשחק בודק איזה קלף גבוה יותר ונותן למנצח את שני הקלפים.

כשהמשחק נגמר (אחרי 7 תורות) המשחק בודק מי ניצח

מציג למסך מי השחקן שניצח ושואל אם רוצים להמשיך ולשחק שוב (כן / לא(.

## תכנון והגדרת המחלקות למשחק

כדי להתחיל לתכנן מחלקות (Classes) ו -UML ים למשחק הקלפים שלנו צריך לגשת לשני היבטים עיקריים :

**תיכנון ותכנות**

כדי לתכנת את המשחק יש להתחיל עם תיכנון המשחק ולשאול את עצמכם, אלו גורמים משתתפים במשחק?

**שלב ראשון: תכנון אובייקטים (OOP)**

נתחיל בזיהוי ה"ישויות" המרכזיות של המשחק. אלה יהפכו למחלקות .

Card - קלף

Deck – חפיסת קלפים

Player - שחקן

Game - משחק

**שלב שני: תכנון UML ים**

לכל אחד מהישויות שהוגדרו יש לבנות Class משלו.

כדי להגדיר כל מחלקה נשאל שתי שאלות:

1. מה התכונות של כל מחלקה איזה מידע היא שומרת? אילו משתנים פנימיים הוא צריך?
2. אלו פעולות הוא יודע לעשות?

כשמגדירים פעולה חושבים על שלוש שאלות:

1. מה שם הפעולה? כדאי לבחור שם המייצג את מה שהפעולה אמורה לבצע.
2. מה הטיפוס שהפעולה מחזירה?
3. מה הפרמטרים שהפעולה מקבלת?

יכול להיות שתוך כדי בניית המשחק נצטרך פעולות נוספות, במידה וכן נוסיף אותם ל UML הבסיסי ונעדכן אותו.

## הגשת התרגיל

כל אחת מהמחלקות בקובץ נפרד

קובץ ראשי main

צילום מסך של הרצת התוכנית

תרשים של uml המחלקות

ניתן בקובץ zip או בקובץ word שאליו יועתקו חלקי התרגיל.

ההגשה דרך ה classroom

לא תתאפשר הארכת זמן בהגשה כיוון שהתרגיל כולל ציון

## המחלקה קלף Class Card

## A screenshot of a computer AI-generated content may be incorrect.

במקרה שלנו:

Enum Shape { Club, Diamond, Heart, Spade}

לכל משתנה בenum יש ערך מספרי, למשתנה הראשון ערך 0.

Shape s = Shape.Diamond;

המשתנה s יקבל את הערך 1

UML בסיסי

|  |  |
| --- | --- |
| Card | |
| קבועים לשימוש | public Enum Shape{Club, Diamond, Heart, Spade} |
| תכונות | private Shape shape  private int value |
| פעולות בונות | public Card(Shape shape, int value) |
| פעולות אחרות  שימו לב: שערך גבוה מסמל אצולה, ההדפסה צריכה להיות בשפת המשתמש | public int getValue()  public int getShape()  public override string ToString() |

שאלה למחשבה: למה לא יצרנו את הפעולה set ?

## המחלקה חפיסה Class Deck



מה תכונות חפיסת הקלפים?

אוסף של עד 52 קלפים (מיוצג בעזרת מערך קלפים(.

כמות הקלפים בפועל - מימוש באמצעות מערך מצוצמם כפי שנלמד בכיתה

למה משמשת חפיסת קלפים?:

מנהל המשחק, חפיסת קלפים לשחקן חפיסת קלפים של ניצחון לשחקן.

מה ההבדל בין החפיסות?

UML בסיסי

|  |  |
| --- | --- |
| Deck | |
| קבועים | public const int MaxDeckSize = 52 |
| תכונות | private Card[] cards  private int deckCount |
| פעולות בונות - בונה חפיסת קלפים ריקה או מלאה | public Deck(bool full) |
| פעולות אחרות  אתחל את החפיסה מחדש  ערבב את החפיסה  משוך קלף מראש החפיסה, הקלף יוצא מהחפיסה  הוסף קלף לחפיסה  שימו לב: הדפיסו רק את הקלפים בחפיסה | public void reset(bool full)  public void shuffle()  public Card pull() public bool add(Card card)  public override string ToString() |

השיטה הנהוגה לערבוב קלפים נקראת **, Fisher-Yates Shuffle** – והיא האלגוריתם הסטנדרטי לערבוב

אקראי של מערך

האלגוריתם:

רוצו בלולאה כגודל המערך

הגרילו מספר בין 0 לגודל המערך

החליפו בין בקלף במיקום הלולאה לקלף במיקום שהוגרל

## המחלקה שחקן Player



מה התכונות של שחקן?

חישבו איך נראות חפיסות הקלפים של השחקן:

בתחילת הסיבוב?

בסוף הסיבוב?

בתחילת סיבוב חדש?

UML בסיסי

|  |  |
| --- | --- |
| Player | |
| תכונות | private string name  private Deck playDeck  private Deck winDeck |
| פעולות בונות | public Player(string name) |
| פעולות אחרות  חלק קלף משחק לשחקן, הוסף לחבילת המשחק  התכונן למשחק חדש  משוך קלף מראש חפיסת המשחק  הוסף קלפים לחבילת הזכיות  שימו לב: הדפיסו רק את הקלפים בחפיסות | public void recieveCard(Card card)  public void reset()  public Card pull() public bool winRound(Card card1, Card card2)  public override string ToString() |

חשבו איזה פעולות חסרות והוסיפו ל uml

## המחלקה משחק Game



תפקידי המשחק:

* ניהול המשחק  
  תמיכה בסיבוב תור בין שחקנים
* הצגת ניקוד השחקנים
* ועוד....

|  |  |
| --- | --- |
| Game | |
| קבועים | private static const int MAXROUNDS = 7; |
| תכונות | private Deck deck  private Player player1  private Player player2  private int current round |
| פעולות בונות | public Game(Player player1, Player player2) |
| פעולות אחרות  מאתחל חפיסה, מאתחל את השחקנים ומחלק להם קלפים ראשונים  מחלק MAXROUNDS קלפים ראשונים לכל שחקן  משחק סיבוב אחד: לוקח קלף מכל שחקן מחזיר את שני הקלפים למנצח (אם קיים)  משחק MAXROUNDS סיבובים מריץ את הסיבובים, מציג את התוצאות, מכריז על מנצח  מאפס את החפיסה ואת השחקנים | public void startGame()  public void dealInitialCards()  public void playRound()  public void playRounds()  public void Run()  public void resetGame()  public override string ToString() |

חשבו האם צריך פעולות נוספות?

מה צריך להופיע בתוכנית הראשית?

שאלות מחשבה:

האם בשינויים קלים תוכלו:

1. שבמשחק יהיו שלושה שחקנים
2. משחק מסוג אחר, כמו רמי, או פוקר?

עבודה נעימה

### נספח 1:

## אוסף המוגדר באמצעות מערך

A close up of a document

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.