**תיאור הפרויקט:**

מדובר במערכת לניהול הופעותיה של תזמורת, והיא מכילה את המחלקות הבאות:

1. תזמורת (Orchestra).
2. הופעה (Concert).
3. פרסונה (Person).
4. נגן כלי נגינה (InstrumentPlayer).
5. מוסיקאי (Musician).
6. מנצח (Conductor).
7. כלי נגינה (Instrument).
8. פסנתר (Piano) .
9. פסנתר אלקטרוני (ElectricPiano).
10. כינור (Violin).
11. צ'לו (Cello).
12. חליל צד (Flute).
13. יצירות (MusicPiece).
14. שעה (Time).
15. תאריך (Date).
16. מחלקת חריגות של התזמורת ((OrchestraExceptions. – ישנן 5 חריגות יורשות ממנה, מצוין בטבלה מטה.
17. רשימה מקושרת כללית בהתאם לטיפוס (GeneralLinkedList).
18. איבר של רשימה מקושרת בהתאם לטיפוס (ListNode).
19. תבנית לשיטת המפעל (OrchestraFactory).
20. מתודות לקליטת נתונים (Utilities).
21. מתודות לשימוש במערכת (ConsoleInterface).

**הורשות:**

|  |  |
| --- | --- |
| **Derives/** **Inherits from:** | **Class Name:** |
| Person, InstrumentPlayer | Musician |
| Person | Conductor |
| Instrument | Piano |
| Piano | ElectricPiano |
| Instrument | Violin |
| Instrument | Cello |
| Instrument | Flute |
| OrchestraExceptions | InvalidNameException |
| OrchestraExceptions | ObjectExistsException |
| OrchestraExceptions | ObjectNotExistsException |
| OrchestraExceptions | MaxCapacityException |
| OrchestraExceptions | NoSuitableValuesException |

**מחלקות אבסטרקטיות:**

|  |
| --- |
| **Class Name:** |
| Person |
| Instrument |
| InstrumentPlayer |

**המערכת מאפשרת לבצע את הפעולות הבאות:**

1. להציג את נתוני התזמורת (ישנן תת-אפשרויות שונות של תצוגה כמו להציג את הסולנים, להציג את רשימת ההופעות ועוד).
2. להוסיף הופעה חדשה / כלי נגינה חדש / מוזיקאי חדש / יצירה חדשה. – ישנם אפשרויות שונות של הוספה עבור כלי הנגינה ועבור ההופעות. כמו כן ישנן הגבלות על ההופעות, לא ניתן לקיים שתי הופעות באותו תאריך ושעה, במידה ומנסים להכניס הופעה כזו תינתן אפשרות עריכה להופעה ללא צורך ביצירת הופעה חדשה.
3. להסיר הופעה.
4. לשייך כלי נגינה למוזיקאי. – ישנם הגבלות, לא ניתן לשייך מוסיקאי יותר מפעם אחת לאותו כלי, לא ניתן לשייך יותר מוסיקאים מכמות הכלים הקיימת ולא ניתן לשייך למוסיקאי יותר כלים מהכמות המותרת.
5. להלחין יצירה חדשה על ידי המנצח.
6. להגדיר מנצח חדש עבור התזמורת.
7. לבדוק את הכוונון של הכלים – לכוון כל כלי בנפרד, אי הלחנה של כלי תגרור הורדה של כמות הכלים הזמינים מאותו סוג.
8. להעניק תואר שחקן סולו לאחד המוזיקאים
9. לנגן הופעה – ניגון הופעה יגרום להעלאת כמות ההופעות(וותק) של המשתתפים.

**נעשה שימוש ב-Design Patterns הבאים:**

1. Singleton
2. Prototype
3. Factory

**תיאור המחלקות והשיטות:**

1. תזמורת (Orchestra).

**משתנים:**

* מחרוזת תווים עבור שם התזמורת.

**סוג המשתנה:** string

* מחרוזת תווים עבור מיקומה הפיזי של התזמורת.

**סוג המשתנה:** char\*

* מספר עבור הכמות המקסימלית של מוזיקאים.

**סוג המשתנה:** int

* מספר עבור הכמות המקסימלית של כלי הנגינה.

**סוג המשתנה:** int

* מספר עבור הכמות המקסימלית של ההופעות המתוכננות.

**סוג המשתנה:** int

* מספר עבור הכמות המקסימלית של היצירות.

**סוג המשתנה:** int

* מספר עבור סך היצירות.

**סוג המשתנה:** int

* מספר עבור סך המוזיקאים.

**סוג המשתנה:** int

* מספר עבור סך כלי הנגינה.

**סוג המשתנה:** int

* מספר עבור סך ההופעות המתוכננות.

**סוג המשתנה:** int

* אובייקט מסוג מנצח (Conductor) המנהל את התזמורת.

**סוג המשתנה:** Conductor

* ווקטור עבור המוזיקאים המשתייכים לתזמורת.

**סוג המשתנה:**  vector<Musician\*>

* ווקטור עבור כלי הנגינה הנמצאים ברשותה של התזמורת.

סוג המשתנה: vector<Instrument\*>

* ווקטור עבור היצירות שהתזמורת יודעת לנגן.

**סוג המשתנה:** vector<MusicPiece\*>

* רשימה מקושרת עבור ההופעות המתוכננות של התזמורת.

**סוג המשתנה:** GeneralLinkedList<Concert>

* **מצביע סטטי לאובייקט מסוג תזמורת.**

**סוג המשתנה :** static Orchestra\*

**שיטות:**

* שיטות get & set עבור שם התזמורת ,מיקומה והמנצח שלה. (המשתנים המייצגים את הכמות הנוכחית של הכלים/הופעות/מוזיקאים יתעדכנו אוטומטית עם הוספה/הורדה מהמערך והמשתנים המייצגים את הכמות המקסימלית יתקבלו בבנאי).
* שיטות add עבור כל המערכים.
* שיטת remove עבור מערך ההופעות. (בהתאם למה שאפרת אמרה ב-17.5).
* שיטת createAutoConcert היוצרות בהתאמה הופעה חדשה בצורה אוטונומית.
* שיטת makeSoloMusician ההופכת מוסיקאי לסולן.
* שיטת assignInstrument המקצה כלי למוסיקאי.
* שיטת addMusicPiecesToAddedConcert המוסיפה יצירות להופעה האחרונה שנוצרה.
* שיטת compose הפונה למנצח לצורך הלחנת יצירה חדשה והוספתה למערך היצירות.
* שיטת checkInstrumentsTuning הבודקת את כל כלי הנגינה ומעדכנת במידה ונדרש טיפול.
* dtor
* שיטות displayעבור הצגת נתוני התזמורת.
* שיטת displayBasicOrchestra המציגה את הנתונים הבסיסים של התזמורת.
* שיטת displaySoloMusicians המציגה את כל הסולנים של התזמורת.
* שיטת displayConcertList המציגה את רשימת ההופעות של התזמורת.
* שיטת displaySingleConcert המציגה את נתוני ההופעה הספציפית בצורה מלאה (כולל את רשימת היצירות שינוגנו בה).
* שיטת displayOrchestraMembers המציגה את המנצח והמוזיקאים שנמצאים בפיקודו.
* שיטת displayAllMusicians המציגה את כל המוזיקאים של התזמורת.

1. הופעה (Concert).

**משתנים:**

* אובייקט מסוג תאריך
* אובייקט מסוג שעה
* מספר עבור כמות היצירות המקסימלית. – מוגבל ל-6 יצירות.

**סוג המשתנה:**  const int

* מספר עבור כמות היצירות הנוכחית.

**סוג המשתנה:** int

* ווקטור עבור היצירות שינגנו בהופעה

**סוג המשתנה:** vector<MusicPiece\*>

**שיטות:**

* שיטות get & set עבור התאריך ושעה
* שיטת add עבור היצירות.
* שיטת removeMusicPiece עבור היצירות.
* שיטת clone .
* dtor
* שיטת calcDuration המחשבת את אורך ההופעה(פלוס מינוס אפסילון) בהתאם לכמות היצירות.
* שיטות printList & printConcertInfoהמדפיסות בהתאמה את רשימת היצירות ונתוני ההופעה הכללים (שעה, תאריך ואורך).

**אופרטורים:**

* == (מקבל הופעה ומחזיר ערך בוליאני אם היא קיימת)
* -- (מעדכן את השעה לאחור)
* ++ (מעדכן את השעה קדימה)

1. פרסונה (Person). – מחלקה אבסטרקטית.

**משתנים:**

* מחרוזת תווים עבור שם.

**סוג המשתנה:** string

* מספר עבור וותק – ישמש עבור כמות ההופעות.

**סוג המשתנה:** int

* מספר עבור ה-ID .

**סוג המשתנה:** int

* מספר סטטי עבור ה-ID .– ישמש בעט יציאת אובייקט חדש.

**סוג המשתנה:** static int

**שיטות:**

* שיטות get & set עבור הוותק השם וה-ID.
* שיטת print וירטואלית

**אופרטורים:**

* ++ (מגדיל את הוותק בשנה אחת).
* == בהתאם ל-ID.

1. נגן כלי נגינה (InstrumentPlayer)

**משתנים:**

* מספר עבור כמות כלי הנגינה שהוא יודע לנגן בהם.

**סוג המשתנה:** int

* מספר קבוע עבור הכמות המקסימלית של כלי הנגינה בהן הוא יוכל לנגן.

**סוג המשתנה:** int

* ווקטור מצבעים עבור כלי הנגינה הנמצאים ברשותה של התזמורת בהן הוא יודע לנגן.

**סוג המשתנה:**  vector<Instrument\*>

**שיטות:**

* שיטת get עבור כמות כלי הנגינה שהוא יודע לנגן בהם והכמות המקסימלית.
* שיטת addInstrumentהמוסיפה כלי חדש לרשימה הכלי של הנגן.
* שיטת displayInstrumentsהמציגה את הכלים שהנגן יודע למגן בהם.
* שיטת doesInstrumentBeenAdded המעדכנת אם ניתן להוסיף כלי.
* שיטת doesInstrumentBeenAdded המעדכנת אם הכלי כבר נמצא ברשותו של הנגן.
* שיטת playInstrument ווירטואלית .

1. מוסיקאי (Musician). **יורש מפרסונה + נגן כלי נגינה**

**משתנים:**

* ערך בוליאני המתאר האם מדובר במוזיקאי סולן.

**סוג המשתנה:** bool

**שיטות:**

* שיטות get & set עבור הערך הבוליאני.
* שיטת clone
* שיטת print

1. מנצח (Conductor).

**משתנים:**

* Enum המתאר את הסגנונות של הנגינה (קלאסי, רומנטי, מודרני, ג'אז)
* ערך enum המתאר את הסגנון

**שיטות:**

* שיטות get & set עבור הסגנון.
* שיטת compose המייצרת יצירה חדשה ומחזירה אותה.
* שיטת print

**אופרטורים:**

* אופרטור השמה =

1. יצירה (MusicPiece).

**משתנים:**

* מחרוזת תווים עבור שם היצרה.

**סוג המשתנה:** string

* מחרות תווים עבור שם המלחין.

**סוג המשתנה:** string

* מספר עבור אורך היצירה בדקות.

**סוג המשתנה:** int

* מספר עבור שנת הוצאה – (עבור יצירות וותיקות לא ניתן לדעת את התאריך המדויק)

**שיטות:**

* שיטות get & set עבור הנתונים הנ"ל.
* שטית clone .
* שיטת print.

**אופרטורים:**

* == (מקבל יצירה ומחזיר ערך בוליאני אם מדובר באותה אחת).
* >> אופרטור הדפסה.

1. כלי נגינה (Instrument).

**משתנים:**

* מחרוזת תווים עבור שם היצרן.

**סוג המשתנה:** string

* מספר עבור הכמות של הכלים מסוג זה.

**סוג המשתנה:** int

* מספר עבור כמות המוסיקאים שמנגנים בכלי מסוג זה (המספר לא יכול להיות גדול מהכמות הכוללת של הכלים).

**סוג המשתנה:** int

* מספר עבור כמות הכלים שזמינים לשימוש.

**סוג המשתנה:** int

* התאריך שבו הכלי עבר טיפול בפעם האחרונה (נניח שבכל פעם שמזמינים בעל מקצוע הוא מטפל בכל כלי הנגינה מאותו הסוג).

**סוג המשתנה:** Date

**שיטות:**

* שיטות get & set עבור שם היצרן, הכמות והחודש.
* שיטת print.
* שיטת tuneRequired המחזירה ערך בוליאני במידה ונדרש לכוון את הכלי.
* שיטת tuneInstrument המכוונת את הכלי.
* שיטת detuneInstrument הגורמת לכלי צורך בכיוון (נעשה שימוש בזמן ניגון הופעה).
* שיטת displayLastTunedDate
* שיטת canAssignMusicians
* שיטת play .

**אופרטורים:**

* -- (מקטין באחד את כמות כלי הנגינה שזמינים לשימוש.)
* ++ (מגדיל באחד את כמות כלי הנגינה שזמינים לשימוש.)

1. שעה (Time).

**משתנים:**

* מספר עבור השעות.

**סוג המשתנה:** int

* מספר עבור הימים.

**סוג המשתנה:** int

**שיטות:**

* שיטות get & set עבור הנתונים הנ"ל.
* שיטת print.

**אופרטורים:**

* == (בהתאם לנתונים).
* >> אופרטור הדפסה.

1. תאריך (Date).

**משתנים:**

* מספר עבור השנה.

**סוג המשתנה:** int

* מספר עבור החודש.

**סוג המשתנה:** int

* מספר עבור היום.

**סוג המשתנה:** int

**שיטות:**

* שיטות get & set עבור הנתונים הנ"ל.
* שיטת print.

**אופרטורים:**

* == (בהתאם לנתונים).
* >> אופרטור הדפסה.

**המחלקות הבאות יורשות מהמחלקה האבסטרקטית Instrument את המשתנים והשיטות, ודורסות את שיטת Play בהתאם לסוגן.**

1. פסנתר (Piano) .

**משתנים:**

* מספר קבוע עבור כמות התווים.

**סוג המשתנה:** const int

* מספר עבור כמות התווים בפועל. (אם הפסנתר לא כוון יתכן ואחד התווים לא ניתן לשימוש).

**סוג המשתנה:** int

* ערך בוליאני העונה האם מדובר בפסנתר כנף.

**סוג המשתנה:** bool

**שיטות:**

* שיטות get & set עבור הנתונים הנ"ל.
* שיטת Print.

1. פסנתר חשמלי (ElectricPiano). – **יורש מפסנתר**

**משתנים:**

* מספר עבור כמות הסגנונות (הכוונה לכך שבפסנתר חשמלי אפשר בלחיצה לשנות את הצליל).

**סוג המשתנה:** int

* מספר המייצג את עוצמת הפסנתר (אם העוצמה לא חזקה מספיק אז לא שומעים).

**סוג המשתנה:** int

* ערך בוליאני העונה האם הפסנתר מחובר לחשמל.

**סוג המשתנה:** bool

**שיטות:**

* שיטות get & set עבור הנתונים הנ"ל.
* שיטת Print.
* שיטת clone.
* שיטות שנדרש לממש בעקבות הירושה.

1. כינור (Violin).

**משתנים:**

* מספר קבוע עבור כמות המיתרים.
* **סוג המשתנה:** const int
* מספר עבור כמות המיתרים בפועל (מאחר והכלי לא תוחזק, יתכן כי אחד המיתרים נקרע ונדרשת תחזוקה).

**סוג המשתנה:** int

**שיטות:**

* שיטות get & set עבור כמות המיתרים בפועל.

1. צ'לו (Cello).

**משתנים:**

* מספר קבוע עבור כמות המיתרים.
* **סוג המשתנה:** const int
* מספר עבור כמות המיתרים בפועל. (מאחר והכלי לא תוחזק, יתכן כי אחד המיתרים נקרע ונדרשת תחזוקה).
* ערך בוליאני העונה האם יש מקל מגביה.

**סוג המשתנה:** bool

**שיטות:**

* שיטות get & set עבור כמות המיתרים בפועל והמקל.
* שיטות שנדרש לממש בעקבות הירושה.

1. חליל צד (Flute).

**משתנים:**

* ערך בוליאני העונה האם יש מקשים (כמו קלרינט).

**סוג המשתנה:** bool

* ערך בוליאני העונה האם החליל סתום. (אם לא מתחזקים את הכלי, יתכן שהחליל נסתם ונדרש טיפול, נקווה שזה לא קרה בזמן ההופעה...)

**סוג המשתנה:** bool

**שיטות:**

* שיטות get & set עבור הערכים הבוליאני.
* שיטות שנדרש לממש בעקבות הירושה.

1. רשימה מקושרת כללית (GeneralLinkedList).

**משתנים:**

* מספר עבור הגודל של הרשימה.

**סוג המשתנה:** int

* מצביע לראש הרשימה.

**סוג המשתנה:** ListNode<T>\*

**שיטות:**

* שיטות remove & add עבור הערכים הבוליאני.
* שיטת getLastObject.
* שיטת getObjectAtIndex.
* שיטת getSize.
* Dtor

**אופרטורים:**

* >> אופרטור הדפסה.

1. איבר של רשימה מקושרת (ListNode).

**משתנים:**

* מספר עבור האינדקס של האיבר.

**סוג המשתנה:** int

* מצביע לאיבר הבאה.

**סוג המשתנה:** ListNode\*

* משתנה של הטיפוס הנתון.

**סוג המשתנה:** T

**אופרטורים:**

* >> אופרטור הדפסה.

1. תבנית לשיטת המפעל (OrchestraFactory).
2. **מתודות:**

**:**

* createInstrument
* createConcert
* createMusician
* createConductor
* createMusicPiece

1. מתודות לקליטת נתונים (Utilities).

**מתודות:**

* cleanBuffer
* receiveAlphaChar
* receiveCharSequence
* receiveIntArray
* checkIfValueExists
* reciveBoolValue
* receiveIndexInRange

הערה – המתודות מוודאות קבלת קלט תקין מהמשתמש, כמו כן המערכת עושה שימוש בחריגות לצורך בדיקות פנימיות.

1. מתודות לשימוש במערכת (ConsoleInterface).
2. **מתודות:**

* Help - מציגה את אפשרויות הבחירה.
* displayOrchestra – מציגה את נתוני התזמורת לפי הבחירה.
* addData – מוסיפה נתונים למערכת.
* addConcert תת פונקציה של addData.
* addSoloMusician הופכת מוסיקאי לסולן.
* playConcert מנגנת הופעה.
* composeMusicPiece מיצירת יצירה חדשה.
* checkAndTuneInstrument בודקת ומכוונת את כלי הנגינה.
* changeConductor משנה את המנצח.
* removeConcert מסירה הופעה.
* assignInstrument מקצה כלי למוסיקאי.
* adjustConcertData תת פונקציה של addConcert.