

תכנות מונחה עצמים**עבודת הגשה 3**

מועד הגשה: 18.8.2024 בשעה 23:50

הוראות הגשה:

1. הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
2. אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
3. הגשה דרך מערכת מודול בלבד. שום עבודה לא מתקבלת במייל.
4. תשובה לכל שאלה מעשית צריכה לכלול קובץ h וקובץ cpp עבור כל מחלקה, וכן קובץ cpp עבור הפונקציה הראשית, וקובץ CMakeLists.txt. את כל הקבצים הללו יש לכווץ לקובץ ZIP אחד, ולהגיש אותו. בנוסף, יש להגיש קובץ WORD או PDF עבור השאלות התיאורטיות.
5. יש להוסיף const בכל מקום שאפשר.
6. מותר להוסיף למחלקות שדות שלא צוינו במפורש בשאלה. למשל, משתנים סטטיים. מותר להוסיף פונקציות (מתודות) למחלקות.
7. הארכות יינתנו אך ורק במקרים חריגים ובצרוף אישורים מתאימים. כמו כן, במקרה של ידע מוקדם חובה ליצור קשר עם המרצה לפני מועד ההגשה.
8. שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה.

שאלה 1: (18 נקודות)

נתונה התוכנית הבאה, ואחריה מספר שאלות.

```
1. #include <iostream>
2. using namespace std;

3. class Base {
4.     public:
5.         Base() {cout << "Base::Base()" << endl;}
6.         virtual ~Base() {cout << "Base::~Base()" << endl;}
7.         virtual void f1() const {cout << "Base::f1()" << endl;}
8.         void f2() const {cout << "Base::f2()" << endl;}
9.         virtual void f3() const {cout << "Base::f3()" << endl;}
10. };

11. class Aba: public Base{
12.     public:
13.         Aba() {cout << "Aba::Aba()" << endl;}
14.         ~Aba() {cout << "Aba::~Aba()" << endl;}
15.         void f1() const {cout << "Aba::f1()" << endl;}
16.         void f2() const {cout << "Aba::f2()" << endl;}
17.         virtual void f4() const {cout << "Aba::f4()" << endl;}
18. };

19. class Ben: public Aba{
20.     public:
21.         Ben() {cout << "Ben::Ben()" << endl;}
22.         ~Ben() {cout << "Ben::~Ben()" << endl;}
23.         virtual void f1() const {cout << "Ben::f1()" << endl;}
24.         void f3() const {cout << "Ben::f3()" << endl;}
25. };
```

```

26. int main() {
27.     Base* p = new ABA();
28.     delete p;
29.     return 0;
30. }
    
```

א. (6 נקודות) מלא את הטבלאות הוירטואליות של המחלקות הנ"ל. חובה לרשום לפני כל מתודה לאיזו מחלקה היא שייכת ע"י הוספת הקידומת ::ClassName לפני כל מתודה.

Base class VTABLE	Aba class VTABLE	Ben class VTABLE

ב. (6 נקודות) תאר בפירוט מה קורה כאשר מריצים את הפונקציה הראשית. מה הפלט? אלו פונקציות מופעלות?

ג. (3 נקודות) מה יקרה אם הפונקציות ההורסות לא יהיו וירטואליות?

ד. (3 נקודות) מחלקה מופשטת (אבסטרקטית) היא מחלקה שלא ניתן ליצור ממנה אובייקטים. מדוע אנו זקוקים למחלקה כזאת?

שאלה 2 : (40 נקודות)

יש לפתור את השאלה על-ידי שימוש נכון בפולימורפיזם ו-RTTI. יש לקבוע את המחלקות, את הקשרים בין המחלקות, ואת השדות והפונקציות של כל מחלקה.

חברה של משאיות מבצעת הובלות ברחבי הארץ. לכל הובלה המאפיינים הבאים:

- **מרחק** (מס' שלם של ק"מ)
- **משקל המטען** (מס' שלם של ק"ג)
- בחברה יש 3 סוגי עובדים: מוסכניקים, נהגים, מנהלים.
- לכל עובד ידועים הנתונים הבאים:
- **שם** (מערך של עד 30 תווים. יש להקצות זיכרון באופן מדויק.)
- **מספר תעודת זהות** (שלם ארוך)
- **ותק** (שלם)
- בנוסף לכך, לכל מוסכניק ידוע מספר שעות נוספות (שלם) שעבד בחודש שחלף. לכל נהג יש מערך של ההובלות שביצע במהלך החודש שחלף.
- המשכורות בחברה מחושבות באופן הבא:
- **עבור מוסכניק:** 6000 שקלים + 500 שקלים לכל שנת ותק + 100 שקלים לכל שעה נוספת.
- **עבור נהג:** 300 שקלים לכל שנת ותק + סכום התשלומים עבור כל ההובלות.
- התשלום עבור הובלה הוא: 100 שקלים + 2 שקלים לכל ק"מ. אם המשקל גדול מ-8000 ולנהג ותק של לפחות 3 שנים אז יש בונוס 200 שקלים עבור ההובלה.
- **עבור מנהל:** 14000 שקלים + 1000 שקלים עבור כל שנת ותק.
- כתוב תכנית המכילה מערך של מצביעים לכל העובדים בחברה. (המערך מכיל עובדים מכל הסוגים.) התכנית מבצעת את הדברים הבאים:
- מאתחלת את המערך של מצביעים לעובדי החברה באמצעות קלט מהמשתמש. (עבור כל נהג יש לקלוט את רשימת ההובלות שלו.)
- קולטת מספר תעודת זהות ופרטי הובלה. מוסיפה את ההובלה למערך של הנהג המתאים. אם אין בחברה נהג בעל מספר תעודת הזהות הזה, תודפס הודעת שגיאה.
- מדפיסה את השמות והמשכורות של כל העובדים.
- מדפיסה את מספר הנסיעות הכולל עם משקל גדול מ-8000 ק"ג.
- מדפיסה את שמות המוסכניקים שעבדו לפחות שעה נוספת אחת.
- משחררת את הזיכרון.

שאלה 3 (42 נקודות)

יש לפתור את השאלה על-ידי שמוש נכון בפולימורפיזם ו-RTTI. עליכם לקבוע את המחלקות, הקשרים בין המחלקות, והשדות והפונקציות של כל מחלקה.

רוצים למחשב מערכת של מידע על עובדים, קורסים ומאמרים באוניברסיטה.
לכל קורס ידועים:

- שם (מערך של עד 20 תווים. יש להקצות זיכרון באופן מדויק).
- מספר שעות שבועיות (שלם)
- מספר סטודנטים (שלם)

לכל מאמר ידוע:

- שם המאמר (מערך של עד 50 תווים. יש להקצות זיכרון באופן מדויק).
באוניברסיטה יש מספר סוגים של עובדים:

- **WORKER** הוא עובד. הפרטים הידועים עליו:

- שם (מערך של עד 20 תווים. יש להקצות זיכרון באופן מדויק).
- ותק (שלם)

- **TEACHER** הוא סוג של WORKER המלמד קורסים. לכל מורה יש מערך של קורסים שאותם הוא מלמד.

- **RESEARCHER** הוא סוג של WORKER המבצע מחקר. לכל חוקר יש מערך של מאמרים שאותם הוא פירסם.

- **LECTURER** הוא גם מלמד קורסים וגם חוקר. לפיכך, הוא גם סוג של TEACHER וגם סוג של RESEARCHER.

המשכורות של העובדים מחושבות באופן הבא:

- עבור WORKER: 6000 שקלים + 400 שקלים לכל שנת ותק.
- עבור TEACHER: 800 שקלים כפול מספר שעות שבועיות של כל הקורסים שהוא מלמד. בנוסף, בונס של 300 שקלים עבור כל קורס עם לפחות 100 סטודנטים. בנוסף, בונס 400 שקלים כפול הוותק.

- עבור RESEARCHER: 5000 שקלים + 3000 שקלים לכל מאמר שפירסם + 1000 שקלים עבור כל שנת ותק.

- עבור LECTURER: 1000 שקלים כפול מספר השעות של כל הקורסים שהוא מלמד. בנוסף, 2000 שקלים לכל מאמר שפרסם. וכן 700 שקלים כפול הוותק.

כתוב תכנית המבצעת את הדברים הבאים:

- מאתחלת באמצעות המשתמש מערך של מצביעים לכל עובדי האוניברסיטה. (מכל הסוגים) האתחול כולל גם את המידע על כל הקורסים והמאמרים.

- מוסיפה לפי קלט של המשתמש מאמר אחד לרשימת המאמרים של אחד העובדים. (העובד יכול להיות או RESEARCHER או LECTURER).

- מדפיסה את המשכורות של כל העובדים מכל הסוגים.

- מדפיסה את השמות של כל הקורסים עם לפחות 100 סטודנטים.

- מדפיסה את כל המאמרים שפורסמו.

- משחררת את הזיכרון.

עבודה פוריה !!!