

(++)

תרגול 8

פולימורפיזם

- מילת המפתח virtual לפני משהו אחרת שאינה מוגדרת סוגים
 שונים יכלו לממש בצורה שונה. זאת ה- derived ולא ירשה לממש זכרה נטול,
 ולממש זכירה של base בעזרת מילה מפתח. ההתחלה היא פונקציה וירטואלית.
 מתבצעת בזמן ריצה.

- הכתיבה: `C::Shape::draw();` כאשר מפורט את המממש `draw` של `Shape`.

- הכתיבה: `Shape S = C;`

ואת: `S.draw();` + אה שייך צה ה- `draw` של `Shape`.
 אם מקרה.

כיוון היו פוינטרים: `Shape *S = C;`
`S->draw();` ← בזמן ריצה, הוא ייבצע

מהו *S ויתחיל מהמאמר שלנו לקרא.

- יש 2 - מן העברה `by reference` או `by pointer` ~~הוא~~ תחום לפני
`draw-ref()` מתייחס לפני `circle`. זאת אף שמעלים `by value` עדיין קובו
 הפני של `shape`.

```
class D: public B {
  using B::func;
  void func(int);
  void func(double);
}
```

```
class B {
  virtual void func(int);
  virtual void func(double);
};
```

אם נבצע את הקריאה: `d.func(2.5);` בוודאי שיהיה `func` של `double`
 תראה דיוקא הפני ה- `Class D void func(int)`. התחשיף קודם ייתפס
 אך הלא `func` אותו וק"נה לפני כל ה- `D`, זה הוא ה- `func`.

* `overloading` ו- `overriding`

- ניתן לפתור את הבעיה ב-2 דרכים: או להוסיף פני `func` נוספת
 שהיא `func(double)` ב- `B`, או להוסיף `using B::func` ^{ניתן} `using` `B::func` `using` `B::func` `using` `B::func`
 ולתת את `func` ב- `B` ומתקיים `scope` של `D`.

Statically bound

```
B: B() {
  func(2);
}
```

statically bound - הפונקציה נקבעת בזמן הקompilation ולא בזמן runtime.
- Statically bound - נקבעת בזמן הקompilation.

Virtual -
inline -
Static -
virtual -

```
class B {
  virtual void func(int=0);
}
```

```
class D: public B {
  void func(int=1);
}
```

```
D* d = new D;
B* b = d;
b->func(); // D::func(0);
d->func(); // D::func(1);
```

```
D* d = new D;
B* b = d;
```

```
void B::func();
void D::func();
```

statically bound -
virtual -
D - הפונקציה b -> func();
B - הפונקציה d -> func();
statically bound -
virtual -
נקבעת בזמן הקompilation.

destructor - הפונקציה אשר נקראת כאשר מוחקת אובייקט.
- B destructor - הפונקציה אשר נקראת כאשר מוחקת אובייקט B.
- D destructor - הפונקציה אשר נקראת כאשר מוחקת אובייקט D.

virtual destructor - compiler general destructor

אנחנו יכולים להגדיר virtual destructor וזה קריטי.

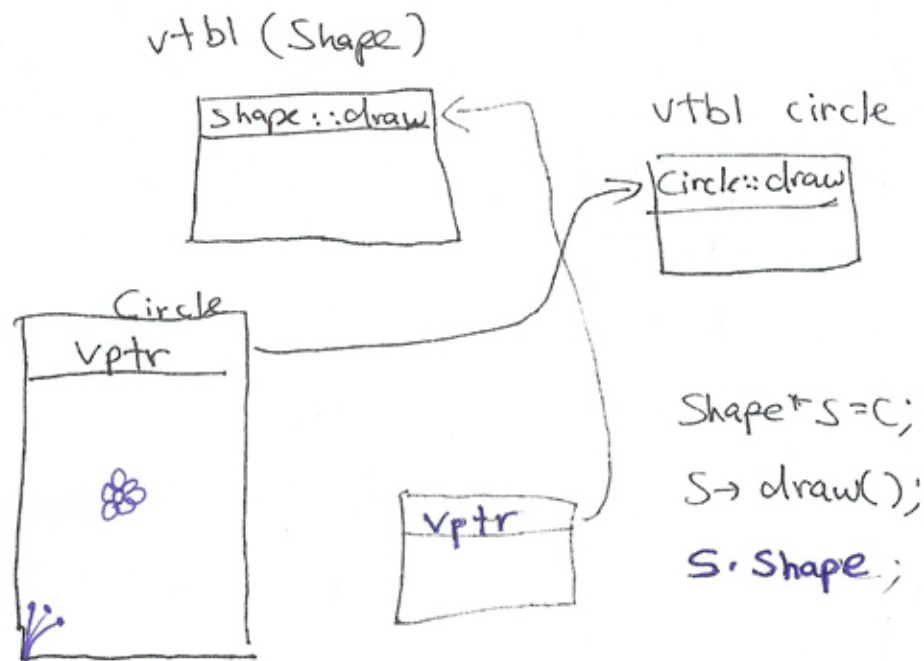
Abstract base class - ABC
base class - ABC
- ABC
- ABC
- ABC

Pure virtual - פונקציה וירטואלית טהורה.

pure virtual - פונקציה וירטואלית טהורה.

- לא ניתן איפיו אנטיק שבו ABC
- אם לא נשלח זרז הפני ה-0, כשנכתב rotate, נקרא בבחינת linking.
- אגב ה-ABC לא יהיה constructor, כי כיון שאם רק נחבר את ה-rotate ל-0, אז ה-rotate יהיה interface.
- ה-destructor ל-ABC מייב אביו (ה-ABC):

אין (אם זמינו) שני לזכרון?



- `vptr` זהו פונקצור שקוונספירי מוסף לא פנו וירטואליזציה.
- גרפויינר הני, זמם ל הפנוי ~~ה~~ בלם `draw` יהיו בעולם ה-0 אם הפנוי בלם `rotate` יהיו בעולם ה-1.



תרגול פ

protected/private ירושה Bella

- ירושה protected/private דומה לירושה.
- ניתן להכיל ~~מחלקה~~ מחלקה עם פרטים. אומות זאת, אין ניתן לבצע ירושה לאותה מחלקה עם פרטים.
- בירושה protected, עבור השתמשו אין ניתן לבטל אבני של מחלקת הבסיס.
- בירושה נג, ~~יש~~ לא public, או private או protected או פרטים public.
- ~~יש~~ פוייטר המחלקה או יכול להכיל את מחלקת הבסיס.

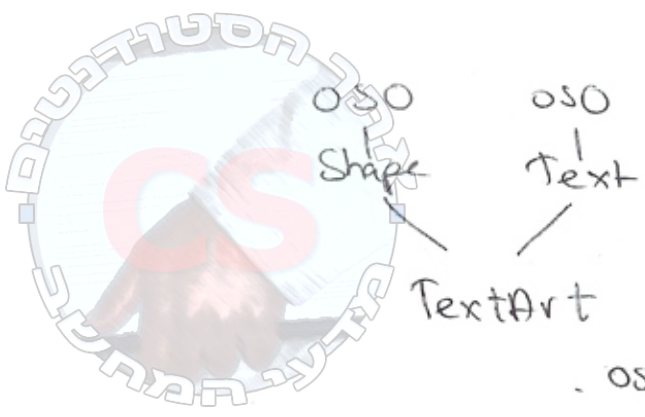
* בירושה protected, גם ה- public וגם ה- protected של מחלקת הבסיס
 עובדים ל- protected.
 בירושה private, גם ה- public, וגם ה- protected
 עובדים ל- private.

* ה- private של מחלקת הבסיס אינם או יעבור בירושה למחלקה derived!
 * הכול אומר: אם ניתן לבצע הברה, רצוי לבצע אותה ולא ירושה פרטית.
שימוש בירושה פרטיג (אומות הברה)

- גישה ל- protected
 - ביצוע overriding למחלקת וירואיות
ירושה מרובה

+ מאפשרת לבצע ירושה מרובה, לומר אומות מחלקה יכולים להיות מספר
 base classes.
 - עם יש לנו ~~אחד~~ פונ' (מחלקת) בסיס שאנו יורשים מהן, ~~יש~~ ניתן כשנרצה
 לבטל לאותה פונ' תהיה שגאת קומפליצה (כי הקומפליצה לא יבד לאינו פונ' לבטל

97A
 יטל אהיוורה מ23:



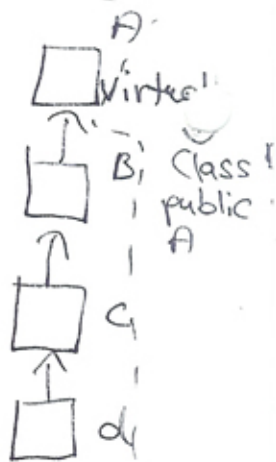
אנו נרצה לזהב את החלק של oso.



מה שנה יקרה, גם ה-Shape וגם ה-text צריכים להפסיק את הירושה מ-Oso במידה virtual.

עמיתו הבסיס base בירושה וירטואלית

באתרו יורשה שאינה וירטואלית, אם האתרו את c, c האתרו את d, ו- b האתרו את a. בירושה וירטואלית, האתרה שממנה יורשים האתרות ה"ע" ב את אחרים. כלומר, d יורשו את c ו-a, c ו-b יורשו את b ו-a. כלומר, כל אלה יורשו את קודמו ואת a.



בני' (2007)



a, d, b, e, c, f
B, C, A, D, E, F, G

בני ה- virtual, צור האתרו:
אם virtual:

