دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر

وراثت و چندریختی — ۶

فایده‌های مقیدسازی پویا

برنامه‌سازی پیشرفته — رامتین خسروی

بهار ۹۹

1

فایده‌ی مقیدسازی پویا چیست؟

2

با مقیدسازی پویا

```
class Vet {
public:
  void give_shot(Animal* a) {
      // do horrible things to a!
      a->make_noise();
  }
};

int main() {
  Dog snoopy;
  Cat garfield;
  Vet ghamar;

  ghamar.give_shot(&snoopy);
  ghamar.give_shot(&garfield);
}
```

با فرض مجازی بودن make_noise

make_noise کلاس Dog صدا می‌شود

make_noise کلاس Cat صدا می‌شود

بدون مقیدسازی پویا

```
class Vet {
public:
  void give_shot_to_dog(Dog* a) {
      // do horrible things to a!
      a->make_noise();
  }
  void give_shot_to_cat(Cat* a) {
      // do horrible things to a!
      a->make_noise();
  }
};
int main() {
  ...
  ghamar.give_shot_to_dog(&snoopy);
  ghamar.give_shot_to_cat(&garfield);
}
```

بدون مقیدسازی پویا

```
class Vet {
public:
  void give_shot_to_dog(Dog* a) {
        // do horrible things to a!
        a->make_noise();
  }
  void give_shot_to_cat(Cat* a) {
        // do horrible things to a!
        a->make_noise();
  }
  // give_shot methods for other animals
};
```

برای هر یک از انواع
حیوانات یک متد جداگانه

بدنه‌ی همه‌ی آنها مشابه است.

با اضافه شدن هر نوع حیوان باید متد مربوط به آن در کلاس Vet اضافه شود.

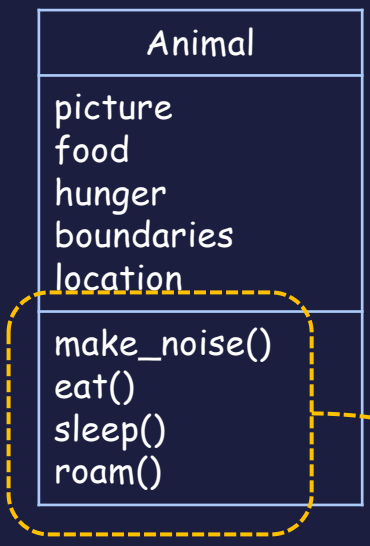
با مقیدسازی پویا ...

حتی برای انواع جدیدی از حیوانات
که بعداً تعریف خواهیم کرد!

Guaranteed to work with
any type of animal!

مطمئن هستیم a->make_noise() کار می‌کند.
چون کلاس a یا make_noise را تعریف کرده یا از Animal به ارث برده.

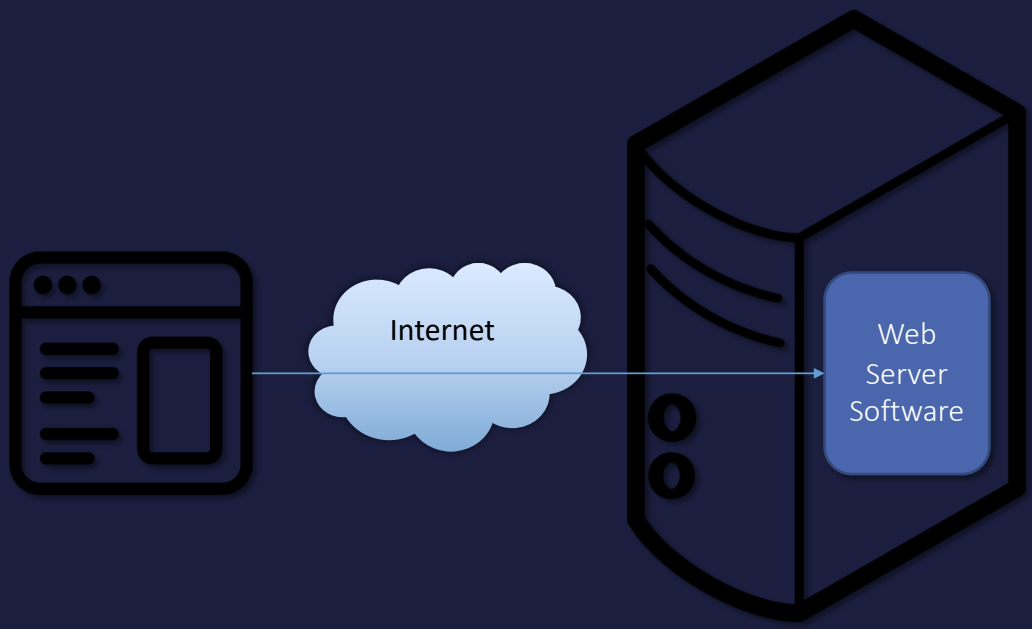
یک قرارداد مشترک برای گروهی از کلاس‌ها




```
class Vet {
public:
  void give_shot(Animal* a) {
    // do horrible things to a!
    a->make_noise();
  }
};
```

متدهایی که مطمئن هستیم تمام
زیرکلاس‌های Animal ارائه می‌کنند
(حتی اگر آن را بازنویسی کنند).

کاربرد دیگر: برنامه‌های وب



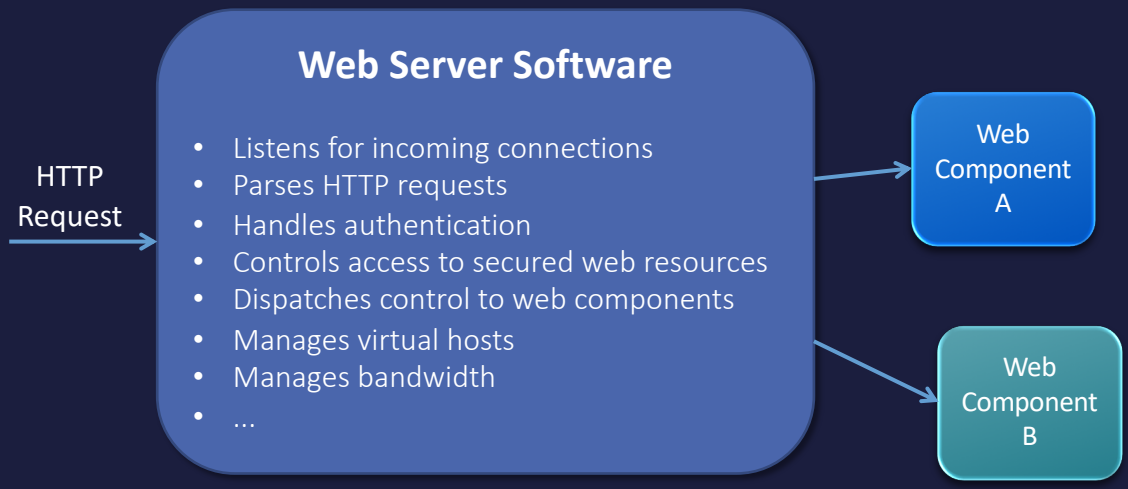
مسئولیت‌های متعدد سرور وب

Web Server Software

- Listens for incoming connections
- Parses HTTP requests
- Handles authentication
- Controls access to secured web resources
- Dispatches control to web components
- Manages virtual hosts
- Manages bandwidth
- ...

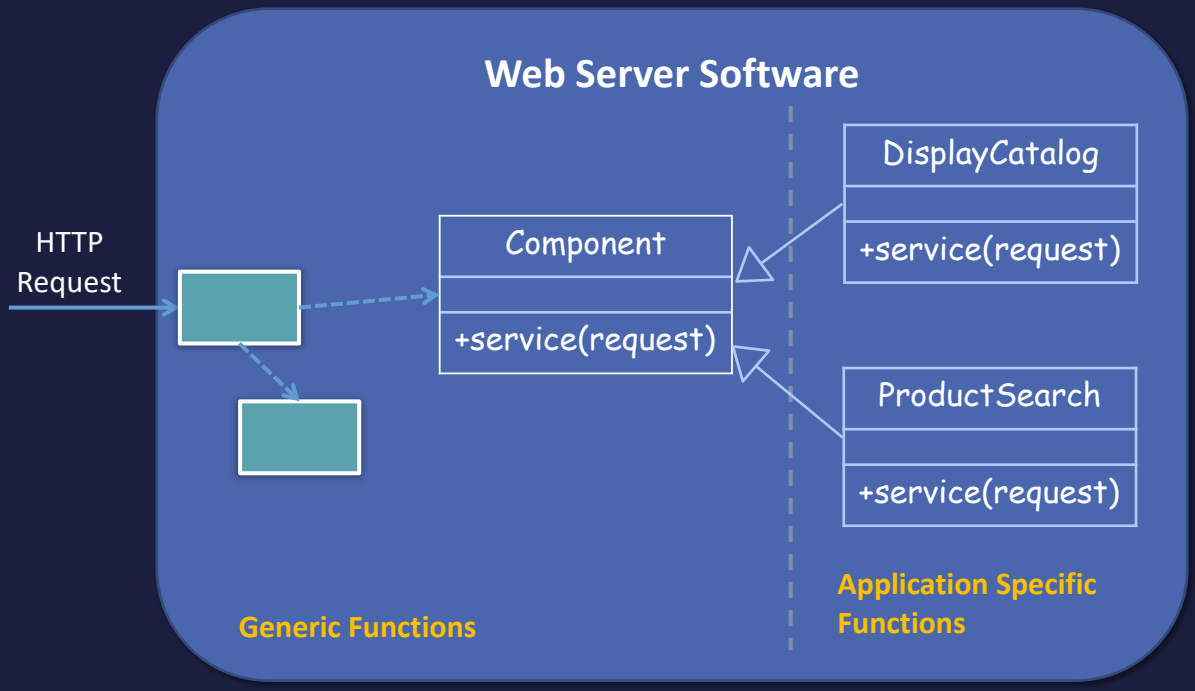
9

مسئولیت‌های متعدد سرور وب



10

مسئولیت‌های متعدد سرور وب



تمرین عملی

تمرین ۱

در برنامه شبیه‌ساز ی توپ و میز به شکلی که در پیوست اسلایدها* در دسترس است یک وابستگی دوطرفه بین کلاس‌های توپ و میز وجود دارد. فرض کنید وابستگی میز به توپ برای ما نامطلوب است (به ویژه این که میز به تمام ویژگی‌های توپ نیازی ندارد). این مشکل را با استفاده از وراثت به این ترتیب حل کنید:

- میز به چه ویژگی‌هایی از توپ علاقه‌مند است؟
- اگر این مجموعه ویژگی‌ها را در یک کلاس جدید قرار دهیم آیا می‌توان برای آن نام معنی‌داری پیدا کرد؟
- اگر پاسخ به سؤال فوق مثبت است کلاسی به این نام تعریف کنید و توپ را از آن مشتق کنید.
- با استفاده از این کلاس، وابستگی میز به توپ را حذف نمایید.

* کد تمرین در Billiards.cpp

تمرین عملی

تمرین ۲

برای هر یک از موارد زیر یک کلاس جدید تعریف کنید (به کمک کلاس‌های تمرین قبل):

- توپ غیرالاستیک: مانند توپ قبل حرکت می‌کند اما برخورد آن با دیواره میز الاستیک نیست، به این ترتیب که پس از هر برخورد با دیواره میز درصدی از انرژی جنبشی آن تلف می‌شود
- پین‌بال: پس از برخورد با دیواره میز در جهتی تصادفی منعکس می‌شود (با حفظ تندی حرکت)
- جوجه اردک منگ: در مسیر مستقیم با سرعت ثابت حرکت می‌کند و هنگامی که به دیواره میز برخورد کرد در همان نقطه برخورد به مدت یک واحد زمان صبر می‌کند و بعد در یک جهت تصادفی به حرکت خود ادامه می‌دهد (با حفظ تندی حرکت)

مشکل‌تر