

مهلت تحویل این تمرین تا پیش از نیمه شب سه شنبه ۱۵/۱۰/۸۳ است. برای ارسال تمرین، یک نامه الکترونیک با مشخصات زیر ارسال کنید:

Recipient:	ramtin@ce.sharif.edu
Subject:	HW6-82112345
Body:	
Attachments:	HW6-82112345.zip

لطفاً مشخصات نامه خود را به طور دقیق تعریف کنید (به کوچکی و بزرگی حروف نیز دقت کنید). فایل HW6-82112345.zip باید شامل متن کلاس های هر دو مسئله ای باشد که در زیر تعریف شده اند.

پوک من!

در این مسئله باید برنامه PooMan که در کلاس درس مورد بررسی قرار گرفت را بهبود بخشید. به این منظور از کلاس های موجود در PooMan.3.zip در شاخه session-25 منابع درس استفاده کنید. در نسخه تغییر یافته برنامه، صفحه اپلت به تعدادی خانه تقسیم می شود طوری که در هر لحظه هر یک از اشیاء بازیکن یا ارواح در یکی از خانه ها هستند. حرکت این اشیاء نیز خانه به خانه انجام می شود و اگر در یک لحظه در یک خانه باشند در لحظه بعد می توانند به یکی از چهار خانه مجاور منتقل شوند. (این حرکت در بازیکن توسط رخدادهای صفحه کلید صورت می گیرد). در صورتی که بازیکن و یک روح با هم برخورد کنند (در یک خانه قرار بگیرند) بازی با نمایش یک پیغام مناسب خاتمه یابد. این ویژگی را طوری پیاده سازی کنید که اجرای همروند ریسمان ها مشکلی در منطق برنامه ایجاد نکند.

ذخیره کننده اشیاء

در این تمرین (که مطابق کوئیز ۱ دی می باشد)، شما باید کلاسی به نام HW51 بنویسید که یک متد با امضای زیر داشته باشد:

```
static String save (InputStream input)
```

ورودی این متد یک «جریان ورودی» می باشد که در آن یک شیء قرار دارد. شما به کمک ObjectInputStream می توانید شیء مربوطه را از جریان ورودی بخوانید. ابتدا باید نوع (نام) این شیء را تشخیص دهید. نوع یک شیء می تواند مانند یکی از موارد زیر باشد:

```
java.lang.String یا oracle.jdbc.driver.OracleDriver
```

این نام شامل نام یک سری package تودرتو می باشد و در انتها نام خود کلاس قرار دارد، که این قسمتها با نقطه از هم جدا شده اند. این متد، باید به ازای هر package یک زیرشاخه بسازد طوری که ساختار مربوط به package ها حفظ شود. در داخلی ترین زیرشاخه باید فایلی با نام خود کلاس ساخته شود و در این فایل محتویات شیء مذکور ذخیره شود. بعنوان مثال فایلهای دو نمونه بالا در شاخه های بصورت زیر قرار می گیرند:

```
java\lang\String یا oracle\jdbc\driver\OracleDriver
```

اگر زیرشاخه‌ها از قبل وجود داشته باشند، شاخه جدیدی تولید نمی‌شود. اگر فایلی با این نام در شاخه مربوطه از قبل وجود داشته باشد، یک عدد به انتهای نام فایل اضافه می‌شود. مثلاً اگر فایلی با نام `String` وجود داشته باشد، باید نام `String1` را انتخاب کند. اگر `String1` هم وجود داشته باشد، `String2` را انتخاب می‌کند و همینطور سعی می‌کند تا زمانی که نام جدید پیدا شود. این نام در نهایت برگردانده (`return`) می‌شود. یک نمونه فراخوانی این متد بصورت زیر می‌باشد:

```
void f (Socket s){  
    System.out.println(HW51.save (s.getInputStream()));  
}
```

راهنمایی: برای ساختن زیرشاخه‌ها از کلاس `File` استفاده کنید. توجه کنید که روش استفاده از متد `mkdir()` که در کلاس `File` تعریف شده، فقط برای سادگی بوده و شما در این تمرین باید فراخوانی صحیح آن را استفاده کنید. برای قسمت‌های مختلف این برنامه می‌توانید به نمونه برنامه‌های موجود در سایت درس مراجعه کنید.

