

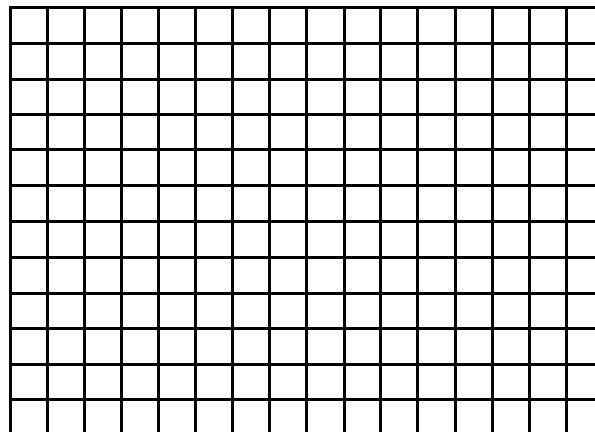
## به نام خداوند مهربان

نکته: با توجه به اینکه در کوئرا برای این سوال فقط یک فایل جاوا می توانید بارگذاری کنید، همه کلاسهایتان را باید در یک فایل قرار دهید (فقط یک کلاس public می توانید در آن فایل داشته باشید که متد main در آن قرار دارد).

### سوال تشریحی کوئرا ۱:

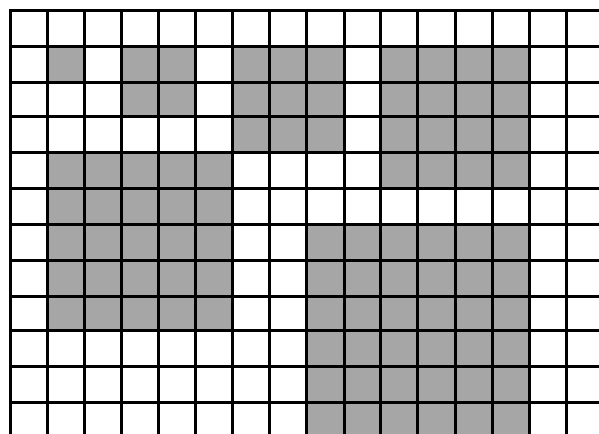
در این سوال می خواهیم برای اشیاء مختلف ابتدا یک طراحی منطقی داشته باشیم و سپس متدهای مورد نیاز را پیاده سازی کرده تا تمامی موارد خواسته شده در زیر به درستی انجام شوند. در این سوال در ابتدا باید برای اشیای مختلف طراحی شیء گرا را انجام دهید. برای گرفتن نمره کامل باید تمامی مباحث خوانده شده را به بهترین نحو استفاده کنید.

در این تمرین یک صفحه مشبک (بورد) داریم که در نهایت اشکال مختلف روی این صفحه رسم خواهند شد. هر خانه از این صفحه نشان دهنده یک پیکسل است. به عنوان مثال تصویر زیر یک صفحه با ۱۲ سطر و ۱۶ ستون را نشان می دهد. شیوه آدرس دهی در این صفحه شبیه چیزی است که در GUI دیدید. یعنی مختصات  $(0, 0)$  نشان دهنده خانه (پیکسل) گوشه بالا سمت چپ صفحه است.

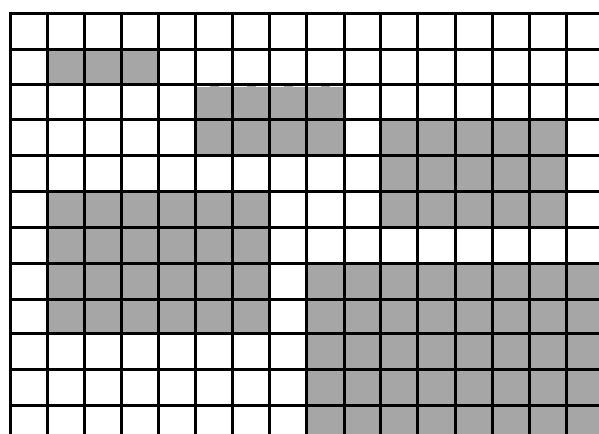


بر روی این صفحه اشکال زیر می توانند قرار گیرند.

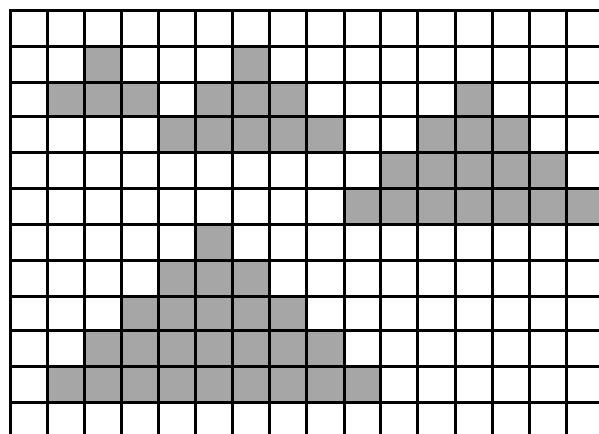
۱) مربع (Square) که دارای یک پارامتر اندازه است که نشان دهنده طول اضلاع آن است. در صفحه زیر ۶ مربع مختلف رسم شده است.



۲) مستطیل (Rectangle) که دارای دو پارامتر است که به ترتیب نشان‌دهنده تعداد سطر و ستون آن است. در شکل زیر تعداد مختلفی مستطیل رسم شده است.



۳) مثلث (Triangle) که دارای یک پارامتر است که نشان‌دهنده اندازه ارتفاع مثلث است و عددی بزرگتر و یا مساوی ۲ است. برای سادگی کار در اینجا فقط مثلث متساوی الساقین داریم که اندازه قاعده آن از روی ارتفاع مشخص می‌شود. در صفحه زیر چندین مثلث مختلف رسم شده است. توجه کنید که فقط الگوی زیر مجاز است، یعنی فقط زاویه ۴۵ درجه مجاز است و تمام مثلث‌ها رو به بالا هستند.



قبل از اینکه ادامه سوال را توضیح دهیم ابتدا یک نمونه تست کیس زیر را در نظر بگیرید:

15 23

4

Square S1 (0,3) 5

Rectangle S2 (3,20) 2 5

Triangle S3 (14,6) 3

Rectangle S4 (7,12) 3 4

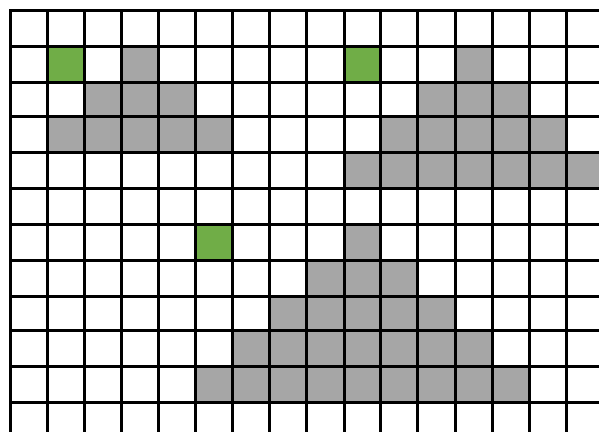
3

IsDrawable

CheckOverlap

Move S4 (10,3)

خط اول تست کیس تعداد سطر و ستون صفحه را نشان می‌دهد که هر دو در یک خط قرار دارند و با یک فاصله از هم جدا شده‌اند. خط دوم تعداد (n) شکلی است که قرار است به صفحه اضافه شوند. n خط بعدی نشان‌دهنده اشکال است که هر خط چندین قسمت دارد که با فاصله از هم جدا شده‌اند. اولین قسمت نوع شکل را نشان می‌دهد. دومین قسمت شناسه شکل است که هر رشته‌ای می‌تواند باشد. سومین قسمت مکان گوشه سمت چپ بالای شکل را نشان می‌دهد که دو عدد است که در پارانتز باز و بسته قرار گرفته و با یک , از هم جدا شده‌اند. برای مربع و مستطیل می‌شود مکان خانه گوشه سمت چپ بالای آنها ولی برای مثلث می‌شود مکان گوشه سمت چپ بالا از مستطیل در برگیرنده شکل مورد نظر. به عنوان مثال، در شکل زیر خانه‌های سبز نشان‌دهنده این خانه برای سه مثلث با اندازه‌های مختلف است.



اولین خط بعد از  $n$  خط (در این مثال ۴ خط) مربوط به اشکال، تعداد دستورهای است که باید اجرا شوند و برای آنها نیاز است خروجی مناسب نیز تولید شود. در تست کیس سه دستور وجود دارد. در کل نیز سه دستور مختلف بیشتر نداریم که در زیر توضیحات آنها را خواهید دید. دستورهای `IsDrawable` و `CheckOverlap` هیچ پارامتری ندارند اما دستور `Move` دو پارامتر مختلف دارد که در قسمت مربوط به این دستور توضیح داده خواهد شد.

مواردی که باید پیاده‌سازی کنید که عملاً می‌شود پیاده‌سازی‌هایی که بتوانید سه نوع دستور بیان شده را اجرا کنید از این قرار هستند که ابتدا باید یک کلاس `Main` داشته باشید که تنها کلاس `public` برنامه است و تنها یک متد `main` دارد که در آن فقط باید چند خط کد جهت ساخت یک شیء از کلاس‌های دیگر و فراخواندن یک و یا چند متد آنها باشد. برای مدیریت خواندن تست کیس‌ها و نوشتن پاسخ‌ها یک کلاس جداگانه بنویسید که فقط این کار را انجام می‌دهد. این کلاس ابتدا کل تست کیس را خوانده و آن را برای پردازش باقی کلاس‌ها آماده می‌کند. در ادامه توضیحات مربوط به سه دستور اصلی برنامه را مشاهده می‌کنید.

۱. اولین دستوری که در تست کیس بالا داشتیم دستور **`IsDrawable`** است. برای این دستور باید متد `isDrawable` از هر کلاس را پیاده‌سازی کنید که بررسی می‌کند آیا تمام شکل در داخل صفحه قرار می‌گیرد یا نه. در صورتی که اینگونه باشد `true` و در غیره این صورت `false` را برمی‌گرداند. در کلاس اصلی که وظیفه مدیریت صفحه را دارد یک متد با اسم `isDrawable` وجود دارد که برای اجرای دستور مربوطه نوشته خواهد شد. در این متد شما باید چک کنید که آیا تمام اشیاء (مثلاً در تست کیس هر ۴ شکل) داخل صفحه قرار می‌گیرند یا نه. در صورتی که اینچنین باشد این متد در خروجی رشته **`true`** را چاپ می‌کند و به خط بعد می‌رود (مثلاً از `println` استفاده کنید تا به خط بعد برود) و در غیر این صورت لیست شناسه اشکالی که از صفحه بیرون زده‌اند را در خروجی چاپ می‌کند که هر شناسه در یک خط مجزا

چاپ خواهد شد و در نهایت به خط بعد می‌رود. دقت شود که لیست شناسه‌ها به صورت مرتب شده باید باشند. به عنوان مثال اگر فرض کنیم S2 و S3 از صفحه بیرون بزنند این متد خروجی زیر را تولید خواهد کرد:

S2

S3

۲. دومین دستور ما `CheckOverlap` است که چک می‌کند آیا اشکال وارد شده با هم هم‌پوشانی دارند یا نه. برای این منظور متدی با نام `checkOverlap` در کلاس اصلی باید بنویسید. در صورتی که هیچ هم‌پوشانی‌ای بین تمام اشکال وجود نداشته باشد این متد رشته **true** را در خروجی چاپ کرده و به خط بعد می‌رود و در غیره این صورت مثل متد `isDrawable` لیست تمام اشکالی که با هم هم‌پوشانی دارند را چاپ خواهد کرد به صورت هر شکل در یک خط و در نهایت به خط بعد می‌رود (مثلا از **println** استفاده کنید تا به خط بعد برود). دقت شود که لیست شناسه‌ها به صورت مرتب شده باید باشند. توجه کنید که کنار هم قرار گرفتن اشکال و یا به عبارت دیگر مماس شدن آنها هم‌پوشانی نیست و حداقل دو شکل باید یک پیکسل مشترک داشته باشند. برای پیاده‌سازی این متد بخش‌هایی که مرتبط با هر شکل می‌شود باید در کلاس مربوطه پیاده‌سازی شده و فقط در اینجا صدا زده شوند. سعی کنید قواعد مربوطه را رعایت کنید.

۳. آخرین دستوری که داریم `Move` است که به طور مشابه باید متد `move` را برای آن در کلاس اصلی پیاده‌سازی کنید. وظیفه این متد این است که یک شیء مشخص را به مکان جدیدی انتقال دهد. دستور `Move` دو پارامتر دارد. اولین پارامتر نشان‌دهنده‌ی شناسه شکلی است که باید جابه‌جا شود. دومین پارامتر نشان‌دهنده مکان جدیدی است که شکل باید به آنجا منتقل شود. در اینجا نیز این نقطه نشان‌دهنده نقطه سمت چپ بالا از اشکال مربع و مستطیل و یا چنین نقطه‌ای در مستطیل در برگیرنده مثلث است. برای اینکه یک شکل را در صفحه جابه‌جا کنید فقط می‌توانید ابتدا به صورت افقی (چپ و یا راست) به اندازه نیاز با گام یک پیکسل حرکت کرده و سپس به اندازه نیاز به صورت عمودی (بالا و یا پایین) با گام یک پیکسل شکل را حرکت دهید. پس دقت کنید مجاز نیستید برای جابه‌جایی از هر نوع گامی استفاده کنید، طول گام‌ها یک و جهت آن ابتدا فقط افقی و سپس فقط عمودی است. خروجی این متد در صورتی که بین شکل مورد نظر در طول مسیر با دیگر اشکال هم‌پوشانی اتفاق نیافتد رشته **true** خواهد بود و بعد از آن به خط بعد می‌رود. در صورتی که هم‌پوشانی‌ای اتفاق بیافتد شناسه اشکال هم‌پوشان شده بدون در نظر گرفتن شکلی که حرکت داده شده

است (شکل مشخص شده در دستور) هر کدام در یک خط چاپ خواهد شد و بعد به خط جدید می‌رود.  
دقت شود که لیست شناسه‌ها به صورت مرتب شده باید باشند.

موفق باشید