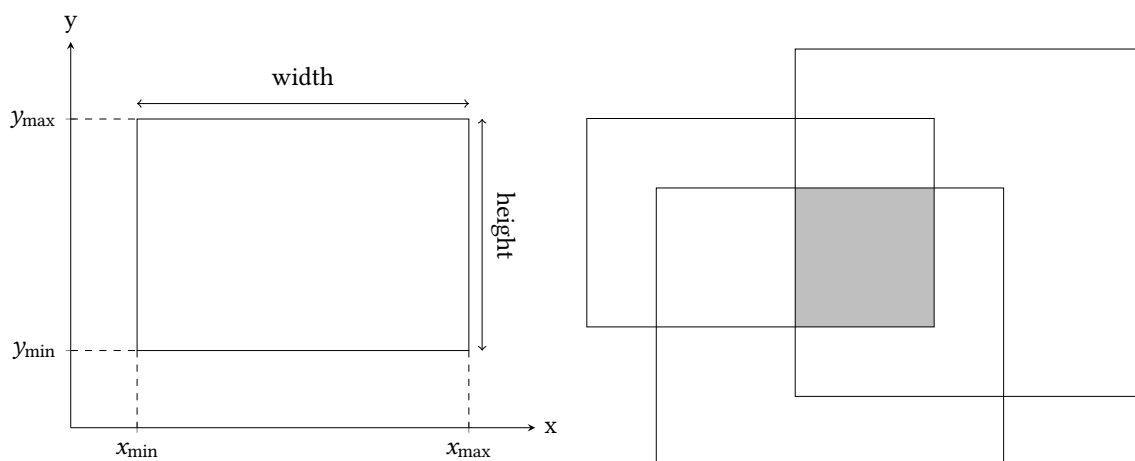


**Allgemeine Hinweise:** Fügen Sie Kommentare hinzu, sobald Sie es für sinnvoll halten. Der Code auf dem main-Branch wird als Abgabe gewertet. Abgaben, bei denen die Kompilierung fehlschlägt, werden mit 0 Punkten bewertet. Wenn Sie bei einer Aufgabe nicht weiterkommen, fragen Sie gerne in den Tutorien oder in Teams nach. Die Designentscheidungen in den Übungsaufgaben entsprechen aus Gründen der Vereinfachung nicht immer denen, die man in der Realität antreffen würde.

## 1 Operationen auf Rechtecken

Diese Aufgabe beschäftigt sich mit zweidimensionalen Rechtecken, deren Seiten parallel zu den Hauptachsen verlaufen. Wir definieren ein solches Rechteck über seine Breite (**width**) und Höhe (**height**) sowie über die Ecke mit den kleinsten Koordinaten (**xmin**, **ymin**). Alternativ kann ein Rechteck über Maxima (**xmax**, **ymax**) und Minima (**xmin**, **ymin**) entlang der Koordinatenachsen festgelegt werden. Die Abbildung unten zeigt ein solches Rechteck und die dazugehörigen Größen. Außerdem ist eine mögliche Operation auf solchen Rechtecken dargestellt: Der Schnitt dreier Rechtecke (grau hinterlegt) ist selbst wieder ein Rechteck.



### 1.1 Rechtecke verstehen (5 P)

Eigentlich sollten Sie die oben genannten Rechtecke jetzt implementieren, aber Ihr Arbeitskollege ist Ihnen zuvorgekommen. Seine Implementierung finden Sie in `src/week04/geometry/Rectangle.scala`. Die Implementierung umfasst die Definition der Klasse **Rectangle** sowie mehrere Operationen, die auf Rechtecken durchgeführt werden können. In der Datei finden Sie außerdem die Methode **makeRectangleFromBounds**, mit der Sie ein Rechteck durch Angabe der Maxima und Minima anlegen können. Außerdem ist eine Konstante für ein leeres Rechteck vordefiniert.

Zusätzlich zur Rechteck-Klasse hat Ihr Kollege in `src/week04/geometry/loops/operations.scala` eine Sammlung von Methoden implementiert, die mit Rechtecken arbeiten. Leider hat Ihr Kollege vorher nur Java programmiert und folglich ist der Code voll mit unleserlichen **while**-Schleifen und haufenweise veränderlichem Zustand. Wenigstens hat Ihr Kollege sich die Zeit genommen, jede Methode zu dokumentieren.

Machen Sie sich mit der Implementierung vertraut und ergänzen Sie die fünf fehlenden Scaladoc-Kommentare in der Klasse **Rectangle**.

## 1.2 Fehler beheben (3 P)

Die Methode `squarePercentage` soll den prozentualen Anteil an Quadraten in der Eingabe ausgeben. Aber einige Tests schlagen fehl.

- (a) Erklären Sie, wo das Problem liegt und warum es auftritt. Schreiben Sie Ihre Erklärung als Kommentar in den Code.
- (b) Passen Sie die Methode so an, dass die Tests nicht mehr fehlschlagen.

## 1.3 Operationen umschreiben (17 P)

In der Vorlesung haben Sie diese Woche verschiedene Transformationsmethoden für Collections kennengelernt.

Schreiben Sie jede der Methoden aus `src/week04/geometry/loops/operations.scala` mithilfe von Transformationen so um, dass **kein veränderlicher Zustand** (`var` und veränderliche Collections), **keine `while`-Schleifen**, **keine `for`-Schleifen**, und **keine explizite Rekursion** verwendet werden, aber dasselbe Ergebnis zurückkommt. Verwenden Sie für diese Aufgabe die Vorlage im Verzeichnis `src/week04/geometry/transformations/`.

Sie dürfen hierfür die Methoden aus der Vorlesung, aber auch jede andere Methode von `ArraySeq` oder `Option` verwenden. Schauen Sie sich hierbei z.B. die Methoden `flatMap`, `reduce/reduceOption`, und `minBy/minByOption` an.

