



# Picasso



# Picasso

- Es soll mit mehreren Threads eine Leinwand (canvas) parallelisiert bemalt werden.



# Aufgabenstellung

1. Jeder erzeugte Thread soll zufällig Felder auf der Leinwand mit passenden Größe auswählen und diese in seiner Farbe (=Index/Nummer/id des Threads) färben. Dies soll aber nur geschehen, wenn das Feld noch nicht gefärbt ist. Wenn auf der Leinwand genug Felder gefärbt wurden (`finished==true`), sollen sich die Threads beenden.

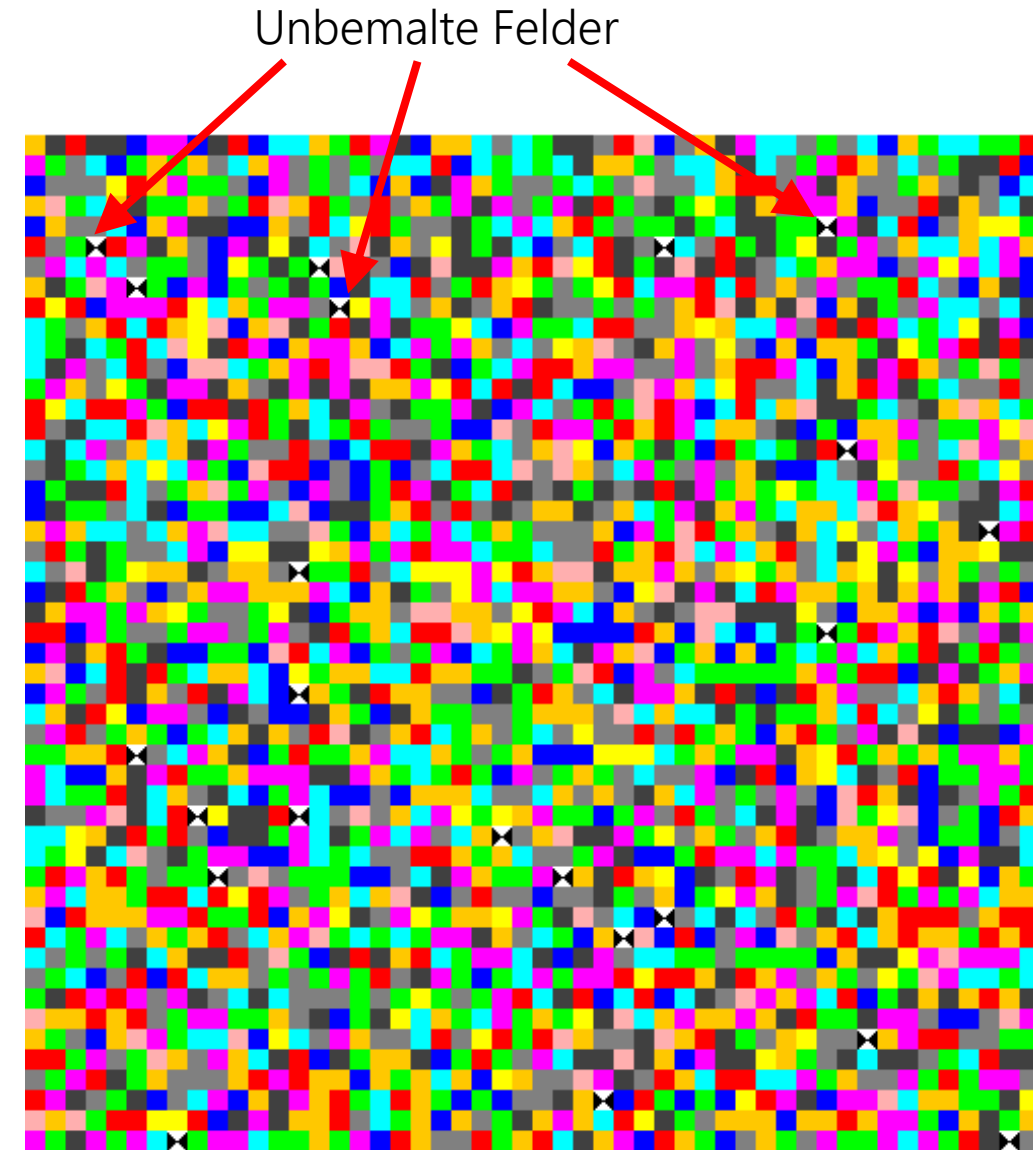
Achte darauf, dass die Feldauswahl der Threads wirklich zufällig ist, und dass jeder Thread beliebig oft jedes Feld auf der Leinwand färben könnte.

# Hilfestellung

- Benutze für die Aufgaben die Methoden der Leinwand `hasColor(...)`, `colorize(...)` und `finished()`. Siehe Klassenkarte der Leinwand!
- Die Zufälligkeit muss mithilfe der Klasse `Random` von Java (`java.util.Random`) erzeugt werden. Einfach ein neues Objekt der Klasse `Random` erzeugen (Standardkonstruktor)  
-> Methode `nextInt(int bound)` für zufällige ganze Zahlen  
Hier darf `Math.random()` nicht genutzt werden, denn dieser Methodenaufruf ist synchronisiert und somit könnte die parallele Implementierung langsamer sein als die sequentielle.

# Wettlaufsituation bei Picasso

- Beim parallelen Bemalen der Leinwand kommt es zu einer Wettlaufsituation, wenn zwei Threads gleichzeitig dasselbe zufällige Feld auswählen – welches noch nicht bemalt ist – und beide dieses Feld mit ihrer Farbe bemalen.
- Dadurch bleibt ein **anderes** Feld unbemalt, weil sich die Leinwand nach Breite\*Höhe-vielen Bemalungen als fertig bemalt ansieht.



# Aufgabenstellung

2. Verhindere mit geeigneten Mitteln diese Wettlaufsituation, sodass die Leinwand immer vollständig bemalt wird!

