# UE Algorithmen und Datenstrukturen (C#)

Abgaben der Beispiele grundsätzlich per Moodle, wie voriges Semester. Beachte jedenfalls die Abgabe- und Bewertungsmodalitäten unter:
`https://wiki.mediacube.at/wiki/index.php/Algorithmen_und_Datenstrukturen_-_SS_2015`.

## 1. Übungsblatt

Abgabe bis 20.03.15 8:55. Inhalt: Command Line Arguments, OOP/Array-Klassen, eigene Exceptions.

1. **Artikel/Store** *(2P)*: ändere die Implementierung von Moodle:
   (a) Verwende eine der beiden C#-Array-Klassen anstatt `Article[] articles`.
   (b) Finde heraus, wie groß die Start-`Capacity` ist, wenn der Konstruktor leer aufgerufen wird.
   (c) Finde heraus, um wie viel die `Capacity` vergrößert wird, wenn ein neuer Artikel in das vollbefüllte Array hinzugefügt wird.
   (d) Dokumentiere alles durch übersichtlichen Code in `Main()`.
2. **RLE** *(5P)*: Komprimiere einen String mithilfe der sogenannten *Lauflängenkodierung*[1].
   (a) Diese Kodierung arbeitet wie folgt: enthält der String mehr als drei identische Zeichen hintereinander, so wird diese Sequenz durch ein einziges Zeichen und die Länge der Sequenz, eingeschlossen in Escape-Zeichen, ersetzt. Es wird also etwa ein `aaaa` zu `a#4#`, wenn `#` das Escape-Zeichen ist (soll der Standard sein). Wenn ein Zeichen nur ein- bis dreimal vorkommt, wird *nichts* verändert.
   (b) Beispiel: aus `AaaaddddDDDEfffxxyZZZZZ` wird `Aaaad#4#DDDEfffxxyZ#5#`, aus `GGGGGGGGGG1111` wird `G#10#1#4#`, aus `ABCDE` wird `ABCDE`.
   (c) Entwickle ein Kommandozeilenprogramm, das nach diesem Schema einen String komprimiert oder dekomprimiert. Das Programm soll mit einem Parameter (`-c` bzw. `-d`) gesteuert werden, ob die Eingabe komprimiert (compress) oder dekomprimiert (decompress) werden soll. Weiters kann mit dem Parameter `-e` das Escape-Zeichen, sofern es anders ist als `#`, angegeben werden.
   (d) In weiterer Folge bezeichnen wir die Executable, die in der Command Line aufgerufen wird, `shrink`. Sie kann auch `main.exe` oder dergleichen heißen. [2]
   (e) Geforderte Funktionsweise:
   der Kommandozeilen-Aufruf `shrink -c AaaaddddDDDEfffxxyZZZZZ` soll die Ausgabe `Aaaad#4#DDDEfffxxyZ#5#` und
   `shrink -d Aaaad#4#DDDEfffxxyZ#5#` wieder die Ausgabe `AaaaddddDDDEfffxxyZZZZZ` liefern.
   (f) `shrink -c 11111 -e *` soll `1*5*` liefern, `shrink -d 1*5* -e *` wieder `11111`.
   Wichtig: *jeder* komprimierte String muss mit `shrink -d` wieder **korrekt** dekomprimiert werden können!
   (g) Teste das Komprimieren und Dekomprimieren **ausgiebig** mit **Debug.Assert**s in separaten Testfunktionen. Achte auch auf kurze Strings, Strings mit identischen Zeichen etc. Es reicht, dass das Programm bei korrekten Aufrufen korrekt arbeitet.
3. **Bonus** (2P): Stelle mithilfe **eigener Exception-Klassen** sicher, dass **jeder mögliche Aufruffehler** gefangen und dem Benutzer mitgeteilt wird.
   ZB: Falsche Parameter; ungültige Anzahl an Parametern; als Escape-Zeichen wird ein Zeichen mitgegeben, das im zu komprimierenden String vorkommt; oder beim zu dekomprimierenden String gibt es das

[1] Auch bekannt als *run length encoding* bzw. *RLE*.
[2] Wird das Programm mit den Skripten auf Moodle kompiliert, wird die Executable nach Ausführung gelöscht. Kommentiere also einfach das Löschkommando in der Skript-Datei aus.

Escape-Zeichen nicht; oder beim zu dekomprimierenden String gibt es das Escape-Zeichen nur einmal etc.

Teste systematisch durch, ob die Exceptions korrekt geworfen werden.

4. **Bonus** *(5P)*: Programmiere die RLE um, so dass sie auf den PNG-Grafiken (binäre Version, P5) des vorigen Semesters[3] funktioniert, und demonstriere dies, indem du ein Bild komprimierst, wieder dekomprimierst, und wieder anzeigen lässt.

5. **Bildgenerierung geometrischer Figuren** *(5P)*

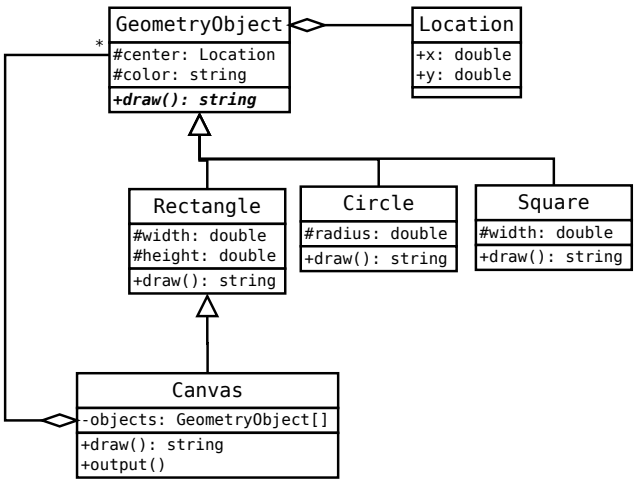
   Schreibe ein Programm, das mithilfe der Software *ImageMagick*[4] [5] einfache Bilder bestehend aus geometrischen Objekten erzeugen kann. Die **deprecated**-Warnung kann ignoriert werden.

   (a) Schreibe dafür zunächst eine Klasse `GeometryObject`, die eine virtuelle Methode `string draw()` hat, die einen leeren string liefert. (Noch besser ist eine abstrakte Klasse mit abstrakter `draw()`-Methode.)

   (b) Ein `GeometryObject` hat weiters eine Farbe (`color`), die als `string` gespeichert wird[6], sowie einen Mittelpunkt `center` vom Typ `Vector`.

   (c) Leite dann von der Klasse `GeometryObject` die Unterklassen `Rectangle`, `Square`, `Circle` und `Canvas` ab:

   - Ein `Rectangle` hat eine Breite (`width`) und Höhe (`height`).
   - Ein `Square` mit nur einer `width`.
   - Ein `Circle` hat einen Radius (`radius`).
   - Die Klasse `Canvas` besitzt ein Array `objects` von `GeometryObject`s.
   - Ein `Canvas` ist von einem `Rectangle` abgeleitet, und hat somit die vererbten Attribute Breite (`width`) und Höhe (`height`).
   - `Canvas` erlaubt das Hinzufügen von Geometry Objekten : mit der Methode `addGeometryObject(GeometryObject object)`, welche das übergebene Objekt zu einem Array [7] von `objects` hinzufügt.




   (d) Die virtuelle Methode `draw()` soll einen leeren String zurückgeben und soll durch die unten angeführten Unterklassen überschrieben werden, sodass folgendes Resultat zurückgeliefert wird:

   - für `Rectangle`: `fill <color> rectangle <x0>,<y0> <x1>,<y1>`
   - für `Circle`: `fill <color> circle <xc>,<yc> <xc + radius>,<yc>`

[3] Alte Angaben, und benötigte Dateien liegen wieder in Moodle vor

[4] `http://www.imagemagick.org/script/binary-releases.php`. Der Link zum Windows-Installer ist `http://www.imagemagick.org/script/binary-releases.php#windows`. Nach der Installation muss ggf. der Rechner neu gebootet werden, damit das benötigte Konsolen-Programm `convert` von der Kommandozeile aus verfügbar ist.

[5] Alternativ steht unter `http://media.zero997.com/convert.php` eine Online-Version von `convert` zur Verfügung

[6] Mögliche Farben sind `"red", "green", "blue", "yellow","black", "white", "transparent"`.

[7] Die Verwendung einer `ArrayList` oder `List<GeometryObject>` anstatt eines Standard-Arrays ist auch erlaubt.

- für `Canvas`: die durch Leerzeichen getrennten Resultate von `draw()` aller `GeometryObjects`, welche sich in im Array `objects` befinden.

Dabei gilt:

- `<color>` ist der Name der Farbe
- `<x0>,<y0>` sind die Koordinaten der linken oberen Ecke
- `<x1>,<y1>` sind die Koordinaten der rechten unteren Ecke
- `<xc>,<yc>` sind die Koordinaten des Mittelpunktes

(e) Implementiere im `Canvas` zusätzlich eine Methode `output()`, die folgenden String ausgibt:

```
convert -size <width>x<height> xc:transparent -draw "<output von draw()
des Canvas>" output.gif
```

(Dieser String kann dann in der Konsole abgesetzt werden, um die Bilddatei `output.gif` zu produzieren.)

(f) Produziere auf diese Weise ein paar verschiedene Output-Bilder und gib sie ebenfalls ab.

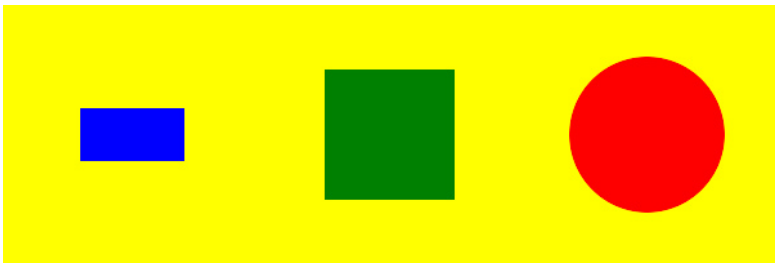
Ein Beispiel, wie die Klassen in `Main()` benützt werden könnten:

```
Canvas canvas = new Canvas(new Location(300,100), "yellow", 600,200);

Rectangle rectangle = new Rectangle(new Location(100,100), 80, 40, "blue");
Square square = new Square(new Location(300,100), 100, "green");
Circle circle = new Circle(new Location(500,100), 60, "red");

canvas.addGeometryObject(rectangle);
canvas.addGeometryObject(square);
canvas.addGeometryObject(circle);

canvas.output();
```

Der Code liefert folgenden (textuellen) Output:

```
convert -size 600x200 xc:transparent -draw "fill yellow rectangle 0,0 600,200
fill blue rectangle 60,80 140,120 fill green rectangle 250,50 350,150 fill red
circle 500,100 560,100" output.gif
```

Und dieser (textuelle) Output wiederum liefert, wenn er als Kommando in der Konsole abgesetzt wird, folgendes Bild `output.gif`:



6. **PGM-Bilder zuschneiden** *(5P)*: Programmiere eine Funktion `byte[][] binarize(byte[][] img)`, welche ein beliebiges Bild (im PGM-Format[8]) nach Schwarz-Weiß konvertiert, sowie `byte[][] crop(byte[][] img)`, welche ein schwarz-weißes Bild so **zuschneidet**, dass komplett weiße Ränder links, rechts, oben, und unten weggeschnitten werden, also im resultierenden Bild nicht mehr vorkommen.

   Besteht das Bild nach `binarize(...)` nur mehr aus weißen Pixeln, soll `crop(...)` ein `null`-Array zurückgeben.

   Testbilder im PGM-Format müssen mitabgegeben werden.

[8] Source Code auf Moodle.