Ergänzungen zu Aufgabe 9

- Mehrere Transformationen sind beteiligt:
  1. Geograph. Daten (Länge, Breite) $\rightarrow$ Kartesische Koord.
     $(\varphi, \psi) \mapsto (x, y, z)$ mit $x = R \cos\psi \cos\phi$
     $y = R \cos\psi \sin\phi$
     $z = R \sin\psi$
     Daten in Grad angegeben, Umrechnung in Bogenmaß erforderlich.
  2. Kartesische Koordinaten $\rightarrow$ Projektionsebene
     $(x, y, z) \mapsto (u, v)$
  3. Projektionsebene $\rightarrow$ SVG-Bildebene (z.B. 500x500 Pixel)
     $(u, v) \mapsto (b_x, b_y)$
     Achtung: y-Achse zeigt nach unten
- Richtige Skalierung wichtig (Orientierungsgröße: Erdumriss)
- Kurven und Kontinentumrisse als Polygonzüge zeichnen - hinreichend viele Unterteilungspunkte verwenden (Ausnahme: Erdumriss)
- Schrittweises Vorgehen sinnvoll: Clipping erst später einbauen

Ergänzungen zu Aufgabe 9 - Fortsetzung

- Bei Erzeugung der SVG-Datei Anführungszeichen im C++-Programm korrekt maskieren.
- SVG-Viewer: inkview (inkscape)
- Informationen zu SVG: http://de.wikipedia.org/wiki/Svg und dort aufgeführte Weblinks
- Sinnvolle Eingabedaten, z.B.: $R = 1$, $D = 5$, $\varphi = 11.5°$, $\psi = 48°$
- Qt5-Programm in mehreren Phasen erstellen: Zunächst nur Graphikfenster anlegen, dann Eingabefenster hinzufügen, zum Schluss Update-Knopf mit update-Funktion.
- Vorgehen ähnlich wie beim Stoppuhr-Beispiel aus der Vorlesung möglich. (Widgets: QHBoxLayout, QSvgWidget, QVBoxLayout, QLabel, QLineEdit, QPushButton)
- Qt Reference Pages: https://doc.qt.io/qt-5/reference-overview.html
- Build: `qmake loes9qt5.pro; make; ./loes9qt5`