



Bremen, Dezember 2025

Aufgabenstellung Entwicklungsarbeit PROG im WiSe 25/26

Hallo zusammen,

die Prüfungsform zum Modul PROG ist die Entwicklungsarbeit, abgekürzt „EA“. Die Prüfungsordnung (Abschnitt II: Prüfungsformen § 7 Arten der Prüfungsleistungen, Studienleistungen) sagt:

„Eine Entwicklungsarbeit besteht in der Erstellung und Demonstration eines Werkes, insbesondere einer Computer-Software oder eines funktionstüchtigen Werkstückes, einschließlich der zugehörigen Dokumentation. Die Dokumentation umfasst in der Regel folgende Dokumente:

- die Aufgabenstellung,
- die Anforderungsdefinition,
- den Entwurf,
- das Quellprogramm,
- die Testdokumentation,
- Benutzungshinweise und
- ein Anwendungsbeispiel.“

Die Entwicklungsarbeit in diesem Modul ist eine Einzelarbeit.

Geben Sie Ihre Entwicklungsarbeit als PDF-Datei, über AULIS spätestens bis zum **01. Februar 2026** ab. Benennen Sie Ihre Datei sinnvoll (PROG_EA_Name_Vorname.pdf). Für die Abgabe finden Sie in AULIS eine Übung ‚Abgabe Entwicklungsarbeit‘ und eine Übungseinheit ‚Abgabe PDF-Datei zur Entwicklungsarbeit‘. AULIS wird Sie 14 Tage vor dem Abgabetermin täglich an die Abgabe erinnern.

Die Demonstration Ihrer Ergebnisse findet im Labor statt. Sie haben im Labor ab Januar 2026 Anwesenheitspflicht. Die Entwicklungsarbeit ist eine sukzessive Prüfungsleistung. Also ein schrittweiser Prozess, der die Reihenfolge und die zeitliche Aufeinanderfolge der Erarbeitung der Ergebnisse betont. Schauen Sie in den Zeitplan, wann welche Teilaufgabe zu demonstrieren ist. Die Dokumentation zur Entwicklungsarbeit wird spätestens am 01. Februar 2026 abgegeben.

Sie dürfen keine grafische Oberfläche (GUI) programmieren. Weder AWT, Swing, JavaFX noch andere GUI-Toolkits sind erlaubt!

Es folgt eine Aufzählung grundsätzlicher Dinge, die Sie in Bezug auf die Bewertung Ihrer Leistung beachten sollen:

- Programmieren Sie objektorientiert. (Objektorientierung)
- Vermeiden Sie überflüssigen, sich wiederholenden, unstrukturierten Code. (Stil)
- Schreiben Sie modulare, überschaubare Methoden. (Modularisierung)
- Setzen Sie Design by Contract ein; kommentieren Sie ihre Methoden mit Vor- und Nachbedingungen. Nur setter- und getter-Methoden können unkommentiert sein. (Design by Contract)
- Schreiben Sie sinnvolle Kommentare und Dokumentationskommentare. Benennen Sie Ihre Bezeichner sinnvoll und verständlich. (Kommentare/Bezeichner)
- Setzen Sie Konstanten ein, da Literale im Code nur als Konstanten verwendet werden sollen. Ausnahmen sind die in der Spezifikation angegebenen Literale wie z.B. ,null‘, ,true‘ und ,false‘, die Zahl 0 als Index und Formeln mit allgemein bekannten Werten, z.B. die 2 bei PI/2. (Literale/Konstanten)
- Halten Sie die Dinge einfach und verständlich. Die Performance steht nicht im Vordergrund. (KISS)
- Erreichen Sie die in den Teilaufgaben geforderten Ziele. (Ziele)
- Strukturieren Sie Ihre Dokumentation geeignet. Halten Sie sich an Grammatik und Rechtschreibung. **Die Dokumentation wird in Deutsch verfasst.** Im Quellcode ist Englisch bei Bezeichnern und Kommentaren erlaubt. Drücken Sie sich kompetent aus und nutzen Sie keine Umgangssprache. Fertigen Sie geeignete Diagramme und Abbildungen (selbst) an. (Form)
- Zeigen Sie Sozialkompetenz in den Vorlesungen, Übungen, praktischen Anwendungen und Demonstrationen der Ergebnisse. (Sozialkompetenz)

Diese Bewertungsbereiche berücksichtigen verschiedene Kompetenzen. Die Fachkompetenz geht mit 50%, die Methodenkompetenz mit 35% und die Sozialkompetenz mit 15% in die Gesamtnote ein. Achten Sie unbedingt darauf, dass die Demonstration der Ergebnisse **und** die Dokumentation in diese Bewertungsbereiche einfließt. Die Demonstration geht mit 50% und die Dokumentation mit 50% in die Gesamtnote ein. Zu jeder Teilaufgabe (siehe unten) gebe ich die erreichbaren Punkte an. In Summe können 100 Punkte von Ihnen erreicht werden. Die Entwicklungsarbeit gilt als bestanden, wenn Sie mindestens 50 Punkte erreichen.

Sie finden die Datei „PROG_Bewertung_Demonstration.pdf“ in der AULIS-Gruppe zu diesem Modul. **Zu jedem Demonstrations-Termin bringen Sie einen Ausdruck dieser Datei mit.** Tragen Sie vor der Demonstration ihren Namen, Vornamen, Matrikelnummer, Studiengang, das Datum und die Nummer der Teilaufgabe ein. Dieser Bewertungsbogen zeigt Ihnen, nach welchen Kriterien die Demonstration bewertet wird.

1. Teilaufgabe (10 Punkte, Taxonomiestufen: Anwenden, Verstehen, Wissen)

Sie finden die Datei 'Windkraftanlagen_DE.csv' in der AULIS-Gruppe zu diesem Modul. Die Daten habe ich von [esri](#). Diese Datei beinhaltet 8909 Datensätze. Jeder Datensatz repräsentiert eine Windkraftanlage.

Jede Zeile dieser Datei enthält die Werte OBJECTID, Name, Baujahr, Gesamtleistung (MW), Anzahl, Typ (WKA), Ort, Landkreis, Breitengrad, Längengrad, Betreiber und Bemerkungen.

Laden Sie die Daten aus der oben genannten Datei ein. Sie dürfen dafür nur die Streams aus der Java API verwenden. Organisieren Sie die Daten in geeigneten Collections. Entwerfen und implementieren Sie die dafür notwendigen Klassen. Das Trennzeichen in der csv-Datei ist das Komma. Geben Sie die Bearbeitungszeit des Einlesens der Datei an. Machen Sie sinnvolle Ausgaben in der Konsole.

2. Teilaufgabe (10 Punkte, Taxonomiestufen: Analysieren, Verstehen, Wissen)

Bei manchen Datensätzen sind die Daten zu Breiten- und/oder Längengrad fehlerhaft. Korrigieren Sie diese Datensätze geeignet (implementieren Sie dafür Methoden in Java). Begründen Sie Ihre Vorgehensweise in den Kommentaren.

Geben Sie die Bearbeitungszeiten des Einlesens der Datei, der Überprüfung und Korrektur der Daten an. Machen Sie sinnvolle Ausgaben.

3. Teilaufgabe (10 Punkte, Taxonomiestufen: Beurteilen, Anwenden)

Sortieren Sie die eingelesenen Daten nach verschiedenen Kriterien. Implementieren Sie dafür Methoden in Java. Begründen Sie Ihre Vorgehensweise in den Kommentaren.

Kombinieren Sie verschiedene Kriterien sinnvoll miteinander. Bewerten Sie in Ihrer Dokumentation den Speicherbedarf und die Dauer der verschiedenen Sortierungen.

Implementieren Sie eine sinnvolle Ausgabe der sortierten Daten. Geben Sie die Bearbeitungszeiten der Sortierung aus.

4. Teilaufgabe (20 Punkte, Taxonomiestufen: Erschaffen, Anwenden)

Implementieren Sie eine Methode, die die südlichste Windkraftanlage sucht.

Implementieren Sie eine Methode, die die Windkraftanlage mit der höchsten Gesamtleistung sucht.

Implementieren Sie eine Methode, die die Windkraftanlage mit den meisten Windrädern sucht.

Implementieren Sie eine Methode, die die Gesamtleistung aller Windkraftanlagen aufsummiert.

Implementieren Sie sinnvolle Ausgaben und geben Sie die Bearbeitungszeiten an.

5. Teilaufgaben (30 Punkte, Taxonomiestufen: Erschaffen, Verstehen, Wissen)

Bei zahlreichen Anlagen fehlen die Daten zur Gesamtleistung. Leiten Sie aus Typ und Standort die fehlenden Angaben ab. Dazu implementieren Sie einen Graphen, mit dem ein Wegenetz der Anlagen aufgebaut wird. Erzeugen Sie für jede Anlage einen Knoten im Graphen. Ist der Abstand zwischen zwei Knoten kleiner oder gleich z.B. 20 Kilometer, tragen Sie eine Kante zwischen den Knoten ein. Wir gehen davon aus, dass im eingestellten Bereich ähnliche Windverhältnisse herrschen. Nutzen Sie den Graphen und Algorithmen, die Sie in Java implementieren, um die fehlenden Daten zu ergänzen.

Um den Abstand zweier Anlagen zu berechnen, verwenden Sie die [Haversine-Formel](#), die die Entfernung zweier Punkte auf einer Kugel unter Verwendung ihrer Längen- und Breitengrade berechnet.

Sie dürfen keine Bibliotheken wie [JGraphT](#) verwenden, sondern sollen den ADT Graph selbst implementieren. Sie haben im Modul 'Grundlagen der Informatik' Graphen als abstrakte Datentypen (ADT) kennen gelernt und wie man diese implementiert. Man verwendet dazu die impliziten Datenstrukturen Adjazenz-Liste oder Adjazenz-Matrix. Bewerten Sie diese beiden Möglichkeiten und entscheiden Sie sich für die für diese Aufgabe geeignete Lösung.

Geben Sie den Graphen geeignet aus und informieren Sie wieder über den Zeitverbrauch.

6. Teilaufgaben (20 Punkte, Taxonomiestufen: Beurteilen, Erschaffen, Analysieren)

Bauen Sie mit dem in Teilaufgabe 5 implementierten Datenstrukturen verschiedene Wegenetze der Anlagen auf. Pro Hersteller von Windkraftanlagen soll ein Wegenetz aufgebaut werden.

Ihr Programm soll einem Wartungsteam ermöglichen, seinen Arbeitsplan zu erstellen. Die Teamleitung wählt eine Anlage aus, bei der gestartet wird. Pro Windrad braucht das Team 2 Stunden für die Wartung. Das Team arbeitet von 8:00 Uhr bis 16:00 Uhr. Die Fahrerin bringt zwischen 16:00 Uhr und maximal 18:00 Uhr das Team zur nächsten Anlage. Wie lange würde das Team brauchen, um alle Anlagen eines Herstellers zu warten?

Geben Sie einen Lösung aus und informieren Sie wieder über den Zeitverbrauch, um diese Lösung zu berechnen.

Beantworten Sie folgende Fragen in Ihrer Dokumentation:

- Welche verschiedene Algorithmen gibt es, um einen Pfad in einem Graphen zu finden?
- Warum möchte man einen kürzesten Weg finden?
- Wie kann man einen kürzesten Weg finden?

Beurteilen Sie die von Ihnen angegebenen Möglichkeiten der Pfadsuche in Ihrer Dokumentation.