



SACHSEN - ANHALT

Ministerium für Bildung

SCHRIFTLICHE ABITURPRÜFUNG 2026

Berufliches Gymnasium Technik

Schwerpunkt: Informationstechnik

Erhöhtes Anforderungsniveau

Variante A

Bearbeitungszeit einschließlich Auswahlzeit: 300 Minuten

Aufgabenkomplex 1: Programmierung

Aufgabenkomplex 2: Netzwerke

Hiermit bestätige ich meine Auswahlentscheidung:

Datum und Unterschrift des Prüflings

Beschreibung der Problemsituation

Moderne Fitness-Apps bieten ihren Nutzerinnen und Nutzern eine Vielzahl an Funktionen zur Analyse und Dokumentation ihrer sportlichen Aktivitäten. Eine zentrale Funktion besteht darin, tägliche Joggingrouten sowie Laufzeiten automatisch aufzuzeichnen. Dabei werden präzise GPS-Koordinaten des Start- und Zielpunkts jeder einzelnen Laufeinheit erfasst – bestehend aus Breitengraden (Latitude, kurz: *lat*) und Längengraden (Longitude, kurz: *lon*).

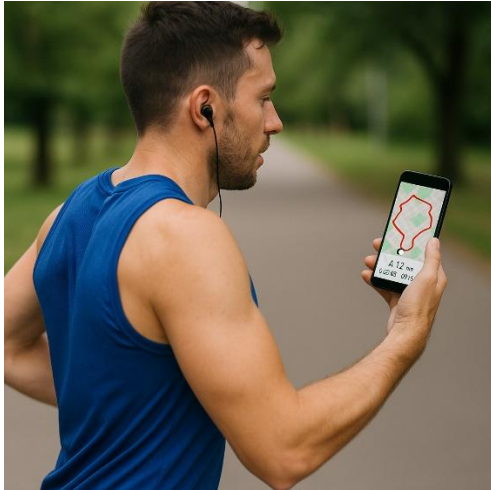


Abbildung 1: Läufer mit Smartphone

Quelle: DALL-E 4, prompted by Stefan Karpe, 25.09.2025

Diese gesammelten Bewegungsdaten werden zur weiteren Verarbeitung und Auswertung auf einem zentralen Server gespeichert.

Was auf den ersten Blick wie ein praktisches Tool zur Selbstoptimierung erscheint, birgt jedoch auch Risiken: Datenschutzexperten warnen, dass sich aus diesen Standortdaten unter Umständen sensible persönliche Informationen ableiten lassen – etwa der regelmäßige Aufenthaltsort, der mit hoher Wahrscheinlichkeit Rückschlüsse auf den Wohnort zulässt, Informationen zum Gesundheitszustand, oder auch wiederkehrende Tagesabläufe, die eine detaillierte Verhaltensanalyse ermöglichen.

Gerade vor dem Hintergrund zunehmender digitaler Vernetzung stellt sich daher die Frage, wie mit solchen sensiblen Daten verantwortungsvoll umgegangen werden kann und welche technischen Maßnahmen geeignet wären, um die Privatsphäre der Nutzer zu schützen, ohne den Nutzen der App einzuschränken.

Aufgabenkomplex 1: Programmierung

Im frühen Stadium der App-Entwicklung stehen zunächst grundlegende Funktionalitäten im Fokus, die später vom Design-Team in eine grafische Oberfläche eingebettet werden sollen. Bereits an dieser Stelle ist es wichtig, auf eine saubere Parametrierung sowie die strukturierte Rückgabe von Daten zu achten, um die spätere Anbindung an die Benutzeroberfläche reibungslos zu ermöglichen.

Ein zentrales Element bildet dabei der Import von Nutzerdaten aus einer Textdatei (*Abbildung 1.1*), in der Informationen zu einzelnen Laufaktivitäten gespeichert sind. Aufbauend auf diesen Daten sollen anschließend verschiedene statistische Auswertungen implementiert werden.

Datum,	Uhrzeit,	Start_lat,	Start_lon,	End_lat,	End_lon,	Strecke_km	Dauer_min
2025-06-01,	20:45,	52.120682,	11.627386,	52.229082,	11.513465,	14.34,	92
2025-06-02,	12:30,	52.120682,	11.627386,	52.150440,	11.508703,	8.75,	56

Abbildung 1.1: Musterdaten-Textdatei

Zu den vorgesehenen statistischen Funktionen zählen:

- die Berechnung der durchschnittlichen Laufzeit in Minuten,
- die Ermittlung der durchschnittlichen Laufstrecke in Kilometern
- sowie die Analyse der maximalen und minimalen Laufgeschwindigkeit in km/h.

Darüber hinaus soll die App Rückschlüsse auf den Biorhythmus der Nutzer ermöglichen. Hierzu wird ausgewertet, wie häufig Läufe in bestimmten Tagesabschnitten stattfinden:

- morgens (05:00 – 10:59 Uhr),
- mittags (11:00 – 14:59 Uhr),
- nachmittags (15:00 – 17:59 Uhr),
- sowie abends (ab 18:00 Uhr).

Hinweise:

Das Archiv „2026_ITE_anlagen_A.zip“ enthält die Dateien „fitnessdaten.txt“ und „laufstrecke.txt“. Beide Dateien müssen für die Funktionalität in denselben Zielordner entpackt werden.

- 1.1** Implementieren Sie eine Funktion zum Einlesen der Daten aus der Textdatei „fitnessdaten.txt“ sowie die Funktionen zur statistischen Auswertung der Laufdaten. Nutzen Sie dabei nicht die vorgefertigten Algorithmen `max` und `min`. Speichern Sie die Lösung unter „Aufgabe11“.
- 25 Punkte**

Hinweis:

Sollte Ihnen das Einlesen der Textdatei nicht gelingen, übernehmen Sie vier Datensätze direkt in den Quelltext.

- 1.2** Implementieren Sie die Funktion zur Ermittlung der Häufigkeiten für den Biorhythmus. Speichern Sie die Lösung unter „Aufgabe12“.
- 8 Punkte**

- 1.3** Diskutieren Sie zwei Lösungen, um Rückschlüsse auf den genauen Wohnort aus den gespeicherten Laufstrecken zu verhindern. Entwickeln Sie eine Lösung wahlweise als Pseudocode, Struktogramm oder in Form eines Quelltextes. Speichern Sie diese Lösung unter „Aufgabe13“.
- 11 Punkte**

Erweiterung der Problemsituation

Im Rahmen der Fitness-App soll die Länge einer einzelnen Laufstrecke (*Abbildung 1.2*) auf Grundlage von GPS-Koordinaten berechnet werden. Dabei werden der Start- und Endpunkt einer Laufeinheit in Form von Breiten- und Längengraden gespeichert.

Da sich die Erde annähernd als Kugel beschreiben lässt, genügt es nicht, einfache lineare Formeln zur Berechnung der Entfernung zu verwenden. Stattdessen kommt die sogenannte Haversine-Formel zur Bestimmung der Distanz d (*Abbildung 1.3*) zum Einsatz – eine mathematische Methode zur Bestimmung der kürzesten Entfernung zwischen zwei Punkten ($P1 (lat1, lon1), P2 (lat2, lon2)$) auf der Oberfläche einer Kugel, also der Luftlinie entlang des Erdumfangs. Die Formel berücksichtigt dabei die Erdkrümmung und rechnet mit dem Erdradius r von ca. 6371 km, um aus den gegebenen Koordinaten die Distanz in Kilometern zu berechnen. Diese Berechnung stellt die Grundlage für viele Anwendungen in der Geolokalisierung dar – von Navigationssystemen bis hin zu Fitness-Trackern.

Punkt	Breitengrad	Längengrad
A (Start/Ziel)	52.120681	11.627386
B	52.124500	11.640000
C	52.135000	11.655000
D	52.129000	11.670000
E	52.118000	11.665000
F	52.115000	11.640000
A (Ziel)	52.120681	11.627386

Abbildung 1.2: Daten einer einzelnen Laufstrecke

$$d = 2 \cdot r \cdot \arcsin \sqrt{\sin^2 \left(\frac{lat2 - lat1}{2} \right) + \cos(lat2) \cdot \cos(lat1) \cdot \sin^2 \left(\frac{lon2 - lon1}{2} \right)}$$

Abbildung 1.3: Haversine-Formel

- 1.4** Implementieren Sie in einer neuen Datei eine Funktion zum Einlesen der Laufstrecke aus der Textdatei „laufstrecke.txt“, eine Funktion zur Bestimmung der Distanz zwischen zwei Punkten unter Verwendung der Haversine-Formel sowie eine Funktion zur Ermittlung der gesamten gelaufenen Strecke. Speichern Sie die Lösung unter „Aufgabe14“. **13 Punkte**

Hinweise:

Sollte Ihnen das Einlesen der Textdatei nicht gelingen, übernehmen Sie vier Datensätze direkt in den Quelltext.

Bei der Umsetzung der Formel können Sie sich an dem Auszug aus der Bibliothek `math` in *Anlage 1.1* orientieren.

- 1.5** Diskutieren Sie, inwieweit die Haversine-Formel in Verbindung mit den gegebenen Daten der Laufstrecke geeignet ist, um die reale Laufstrecke zu ermitteln. Gehen Sie auf notwendige Änderungen ein. **3 Punkte**

Anlage 1.1: Auszug aus der Bibliothek `math`

Funktion	Beschreibung
<code>ceil(x)</code>	Das kleinste ganzzahlige Ergebnis, das größer oder gleich x ist
<code>floor(x)</code>	Das größte ganzzahlige Ergebnis, das kleiner oder gleich x ist
<code>trunc(x)</code>	Der ganzzahlige Anteil von x (Nachkommastellen werden abgeschnitten)
<code>exp2(x)</code>	2 hoch x
<code>pow(x, y)</code>	x hoch y
<code>sqrt(x)</code>	Quadratwurzel von x
<code>degrees(x)</code>	Wandelt den Winkel x von Bogenmaß in Gradmaß um
<code>radians(x)</code>	Wandelt den Winkel x von Gradmaß in Bogenmaß um
<code>acos(x)</code>	Arkuskosinus von x, Rückgabe im Bogenmaß
<code>asin(x)</code>	Arkussinus von x, Rückgabe im Bogenmaß
<code>atan(x)</code>	Arkustangens von x, Rückgabe im Bogenmaß
<code>atan2(y, x)</code>	Arkustangens von (y / x), Rückgabe im Bogenmaß
<code>cos(x)</code>	Kosinus von x in Bogenmaß
<code>sin(x)</code>	Sinus von x in Bogenmaß
<code>tan(x)</code>	Tangens von x in Bogenmaß

Quelle: Vgl. <https://docs.python.org/3/library/math.html>

Aufgabenkomplex 2: Netzwerke

Die Betreiber der Fitness-App planen, ihre Anwendung am 06.06.2026 im Rahmen der Fitnessmesse der Öffentlichkeit vorzustellen. Voraussetzung für die Präsentation ist eine vollständig funktionsfähige App sowie eine betriebsbereite Netzwerkinfrastruktur, über die die erfassten App-Daten zuverlässig an die Server des Anbieters übertragen werden können. Aufgrund krankheitsbedingter Ausfälle kann mit der Erstellung des Netzwerkkonzeptes erst am 04.05.2026 begonnen werden. Daher besteht Unsicherheit, ob der ursprünglich festgelegte Zeitplan eingehalten und das Projektziel termingerecht erreicht werden kann. Die erforderlichen Arbeitsschritte zur Entwicklung und Testsimulation der App wurden bereits dokumentiert (vgl. Anlage 2.1).

Die Simulation der Kommunikationsprozesse zwischen der Fitness-App und den Servern der Betreiber soll mit der Software „Filius“ umgesetzt werden. Ziel ist die Nachbildung einer funktionsfähigen Netzwerkumgebung, die den Datenaustausch zwischen den Endgeräten und den zentralen Diensten des Anbieters realitätsnah abbildet.

Folgendes Szenario soll eingerichtet werden:

- Ein Rechner mit der IP-Adresse 8.8.8.8 /8 soll als öffentlicher DNS-Server unter der Domäne „dns.de“ betrieben werden. Dieser ist über einen Router angebunden, der die Funktion eines Internet Service Providers (ISP) übernimmt und der die Verbindung zu den weiteren Routern herstellt. Der DNS-Server soll alle notwendigen Zoneneinträge für das Netzwerk „fitnessapp.de“ enthalten und gleichzeitig die Namensauflösung für die Smartphones übernehmen.
- Das Betreiber-Netz der Domäne „fitnessapp.de“ soll über einen eigenen ISP-Router mit den anderen Netzwerken verbunden sein. Innerhalb dieses Netzes sollen zwei Server eingerichtet werden: „server.fitnessapp.de“, auf dem ein Echo-Server zur Simulation betrieben wird und „dns.fitnessapp.de“, der als interner DNS-Server fungiert.
- Auf der Nutzerseite sollen zwei Smartphones als Clients integriert werden: „Smartphone 1“ mit der IP-Adresse 192.168.0.10 /24 und „Smartphone 2“ mit der IP-Adresse 172.16.16.10 /24. Beide Smartphones sollen jeweils über einen eigenen Router mit ISP-Funktion an das Gesamtnetz angebunden werden. Auf den Geräten soll ein Client-Programm installiert werden, das die App simuliert und Datenpakete (z. B. GPS-Koordinaten) an den Server „server.fitnessapp.de“ überträgt.

Da bei der Nutzung der Fitness-App personenbezogene Daten verarbeitet werden, müssen auch datenschutzrechtliche Aspekte berücksichtigt werden. Insbesondere ist zu prüfen, welche Arten personenbezogener Daten erfasst werden, welche Risiken für die Privatsphäre der Nutzer daraus entstehen und wie das Prinzip der Datensparsamkeit in der App-Architektur umgesetzt werden kann.

- | | | |
|------------|--|------------------|
| 2.1 | Erläutern Sie anhand dreier Beispiele, welche personenbezogenen Daten eine Fitness-App typischerweise erfasst. Gehen Sie jeweils auf konkrete Datenschutzrisiken ein. | 6 Punkte |
| 2.2 | Beschreiben Sie den Begriff „Datensparsamkeit“ im Zusammenhang mit IT-Systemen. Erklären Sie zwei Möglichkeiten diese Fitness-App datensparsamer zu gestalten. | 6 Punkte |
| 2.3 | Überführen Sie das Netzwerk in die Software „Filius“ und testen Sie die Datenübertragung von GPS-Koordinaten. Speichern Sie die Simulation unter „Aufgabe23.flx“. | 22 Punkte |
| 2.4 | Erstellen Sie anhand der Vorgangsliste aus <i>Anlage 2.1</i> einen Netzplan. Überprüfen Sie, ob das vorgegebene Terminziel erreicht werden kann, und geben Sie das Datum des voraussichtlichen Projektendes an. Orientieren Sie sich dabei am Kalenderauszug in <i>Anlage 2.2</i> . | 14 Punkte |
| 2.5 | Diskutieren Sie, ob und unter welchen Bedingungen es sinnvoll ist, Krankenkassen den Zugriff auf die Daten einer Fitness-App zu ermöglichen. Gehen Sie dabei sowohl auf Chancen als auch auf Risiken ein und beziehen Sie rechtliche sowie ethische Aspekte in Ihre Argumentation ein. | 12 Punkte |

Anlage 2.1: Vorgangsliste

Nr.	Vorgang	Dauer	Vorgänger
1	Bedarfsermittlung & Netzwerkkonzept erstellen (Topologie, Dienste, Sicherheitskonzept)	3 Tage	–
2	Beschaffung der Hardware (Server, Firewall, Switches, Router)	5 Tage	1
3	Lieferung und physische Installation der Geräte im Serverraum (Rack, Verkabelung, Stromversorgung)	2 Tage	2
4	Grundkonfiguration der Netzwerkgeräte (IP-Vergabe, Routing, Firewall-Basisregeln)	2 Tage	3
5	Installation des Betriebssystems auf dem Hauptserver	1 Tag	3
6	Einrichtung des internen DNS-Servers (Domain <i>fitnessapp.de</i>)	2 Tage	5
7	Konfiguration des Echo-Servers (zur Testkommunikation)	1 Tag	5
8	Erweiterte Firewall-Regeln implementieren (Filter, Portfreigaben)	2 Tage	4
9	Integration externer DNS-Einträge (Delegation zu <i>dns.de</i> , IP 8.8.8.8 /8)	1 Tag	6
10	Verbindungstest mit internen Clients (Namensauflösung, Ping, Echo-Test)	1 Tag	6,7,8
11	Simulation der App-Datenübertragung (Textdateien vom Smartphone-Client zum Server)	2 Tage	10
12	Abschlussprüfung und Dokumentation (Netzplan, Testprotokolle, Sicherheitshinweise)	2 Tage	9,11

Anlage 2.2: Kalenderauszug

Mai							Juni						
M	D	M	D	F	S	S	M	D	M	D	F	S	S
				1	2	3	1	2	3	4	5	6	7
4	5	6	7	8	9	10	8	9	10	11	12	13	14
11	12	13	14	15	16	17	15	16	17	18	19	20	21
18	19	20	21	22	23	24	22	23	24	25	26	27	28
25	26	27	28	29	30	31	29	30					

Hinweis:

Wochenenden und Feiertage sind „Fett“ gedruckt und haben eine blaue Hintergrundfarbe.