

```

import csv
import os

# --- Klasser ---
class Person:
    def __init__(self, navn, alder, køn):
        self.navn = navn
        self.alder = alder
        self.køn = køn

    def __str__(self):
        return f"Navn: {self.navn}, Alder: {self.alder}, Køn: {self.køn}"

    @property
    def alder(self) -> int:
        return self._alder

    @alder.setter
    def alder(self, value):
        try:
            value = int(value)
        except (TypeError, ValueError):
            raise TypeError("Alder skal være et heltal") from None
        if value < 0:
            raise ValueError("Alder kan ikke være negativ")
        self._alder = value

class Elev(Person):
    def __init__(self, navn, alder, køn, skole, klassetrin):
        super().__init__(navn, alder, køn)
        self.skole = skole
        self.klassetrin = klassetrin

    def __str__(self):
        return f"{super().__str__()}, Skole: {self.skole}, Klassetrin: {self.klassetrin}"

# --- Filnavn ---
FILENAME = "personliste.csv"

# --- Gem listen til CSV ---
def gem_personer_csv(personer):
    # Find mappen hvor .py filen ligger
    script_dir = os.path.dirname(os.path.abspath(__file__))
    # Kombiner med filnavnet
    filepath = os.path.join(script_dir, FILENAME)

    felt_navn = ["navn", "alder", "køn", "skole", "klassetrin"]
    with open(filepath, "w", newline="", encoding="utf-8") as f:
        writer = csv.DictWriter(f, fieldnames=felt_navn)
        writer.writeheader()
        for p in personer:
            row = {
                "navn": p.navn,
                "alder": p.alder,
                "køn": p.køn,
                "skole": getattr(p, "skole", ""),
                "klassetrin": getattr(p, "klassetrin", "")
            }
            writer.writerow(row)
    print(f"Listen er gemt i '{filepath}' (CSV-fil).")

# --- Indlæs liste fra CSV ---
def indlaes_personer_csv():
    # Find mappen hvor .py filen ligger
    script_dir = os.path.dirname(os.path.abspath(__file__))
    # Kombiner med filnavnet
    filepath = os.path.join(script_dir, FILENAME)

    personer = []
    if os.path.exists(filepath):
        with open(filepath, "r", newline="", encoding="utf-8") as f:
            reader = csv.DictReader(f)
            for row in reader:
                navn = row["navn"]
                alder = row["alder"]
                køn = row["køn"]
                skole = row["skole"]
                klassetrin = row["klassetrin"]
                elev = Elev(navn, alder, køn, skole, klassetrin)
                personer.append(elev)
    else:
        print("Ingen tidligere fil fundet, starter med tom liste")
    return personer

# --- Terminalprogram ---
def main():
    personer = indlaes_personer_csv() # indlæs eksisterende CSV-fil

    while True:
        print("\n--- Person/Elev Registrering ---")
        print("1. Tilføj person")
        print("2. Vis alle personer")
        print("3. Tilføj person til skole")
        print("4. Gem liste som CSV")
        print("5. Afslut")
        valg = input("Vælg en mulighed: ")

        if valg == "1":
            navn = input("Indtast navn: ")
            alder = input("Indtast alder: ")
            køn = input("Indtast køn: ")
            try:
                elev = Elev(navn, alder, køn)
                personer.append(elev)
                print("Person tilføjet!")
            except ValueError:
                print("⚠ Alder skal være et heltal.")

        elif valg == "2":
            if not personer:
                print("Ingen personer registreret endnu.")
            else:
                print("\n--- Registrerede personer/elev ---")
                for i, person in enumerate(personer, start=1):
                    print(f"{i}. {person}")

        elif valg == "3":
            ikke_elever = [p for p in personer if not isinstance(p, Elev)]
            if not ikke_elever:
                print("Ingen personer at opgradere.")
                continue

            print("\nVælg en person at opgradere til elev:")
            for i, person in enumerate(ikke_elever, start=1):
                print(f"{i}. {person}")

            try:
                valg_index = int(input("Nummer: ")) - 1
                person_valgt = ikke_elever[valg_index]
            except (ValueError, IndexError):
                print("Ugyldigt valg.")
                continue

            skole = input("Indtast skole: ")
            klassetrin = input("Indtast klassetrin: ")
            elev = Elev(person_valgt.navn, person_valgt.alder, person_valgt.køn, skole, klassetrin)
            personer[personer.index(person_valgt)] = elev

            print(f"{elev.navn} er nu elev på {skole}, klassetrin: {elev.klassetrin}")

        elif valg == "4":
            gem_personer_csv(personer)

        elif valg == "5":
            print("Program afsluttes.")
            gem_personer_csv(personer)
            break

        else:
            print("Ugyldigt valg.")

```

```

        alder = int(row["alder"])
        køn = row["køn"]
        skole = row.get("skole", "")
        klassetrin = row.get("klassetrin", "")
        if skole or klassetrin:
            personer.append(Elev(navn, alder, køn, skole, klassetrin))
        else:
            personer.append(Person(navn, alder, køn))
    print(f"{len(personer)} personer/elev indlæst fra '{filnavn}'")

    else:
        print("Ingen tidligere fil fundet, starter med tom liste")
    return personer

# --- Terminalprogram ---
def main():
    personer = indlaes_personer_csv() # indlæs eksisterende CSV-fil

    while True:
        print("\n--- Person/Elev Registrering ---")
        print("1. Tilføj person")
        print("2. Vis alle personer")
        print("3. Tilføj person til skole")
        print("4. Gem liste som CSV")
        print("5. Afslut")
        valg = input("Vælg en mulighed: ")

        if valg == "1":
            navn = input("Indtast navn: ")
            alder = input("Indtast alder: ")
            køn = input("Indtast køn: ")
            try:
                elev = Elev(navn, alder, køn)
                personer.append(p)
                print("Person tilføjet!")
            except ValueError:
                print("⚠ Alder skal være et heltal.")

        elif valg == "2":
            if not personer:
                print("Ingen personer registreret endnu.")
            else:
                print("\n--- Registrerede personer/elev ---")
                for i, person in enumerate(personer, start=1):
                    print(f"{i}. {person}")

        elif valg == "3":
            ikke_elever = [p for p in personer if not isinstance(p, Elev)]
            if not ikke_elever:
                print("Ingen personer at opgradere.")
                continue

            print("\nVælg en person at opgradere til elev:")
            for i, person in enumerate(ikke_elever, start=1):
                print(f"{i}. {person}")

            try:
                valg_index = int(input("Nummer: ")) - 1
                person_valgt = ikke_elever[valg_index]
            except (ValueError, IndexError):
                print("Ugyldigt valg.")
                continue

            skole = input("Indtast skole: ")
            klassetrin = input("Indtast klassetrin: ")
            elev = Elev(person_valgt.navn, person_valgt.alder, person_valgt.køn, skole, klassetrin)
            personer[personer.index(person_valgt)] = elev

            print(f"{elev.navn} er nu elev på {skole}, klassetrin: {elev.klassetrin}")

        elif valg == "4":
            gem_personer_csv(personer)

        elif valg == "5":
            print("Program afsluttes.")
            gem_personer_csv(personer)
            break

        else:
            print("Ugyldigt valg.")

```

```
print("Ugyldigt valg, prøv igen.")

if __name__ == "__main__":
    main()
```