# სტუდენტის კლასის განსაზღვრა

class Student:

def \_\_init\_\_(self, name, roll\_number, grade):

self.name = name

self.roll\_number = roll\_number

self.grade = grade

def display\_details(self):

# სტუდენტის დეტალების ჩვენება

return f"სახელი: {self.name}, ნომერი: {self.roll\_number}, შეფასება: {self.grade}"

# სტუდენტების მართვის სისტემა

class StudentManagementSystem:

def \_\_init\_\_(self):

self.students = []

def add\_student(self, name, roll\_number, grade):

# ახალი სტუდენტის დამატება

new\_student = Student(name, roll\_number, grade)

self.students.append(new\_student)

print(f"სტუდენტი {name} წარმატებით დაემატა.")

def display\_all\_students(self):

# ყველა სტუდენტის ჩვენება

if not self.students:

print("სტუდენტების სია ცარიელია.")

else:

print("სტუდენტების სია:")

for student in self.students:

print(student.display\_details())

def remove\_student(self, roll\_number):

# სტუდენტის წაშლა ნომრის მიხედვით

for student in self.students:

if student.roll\_number == roll\_number:

self.students.remove(student)

print(f"სტუდენტი ნომრით {roll\_number} წარმატებით წაიშალა.")

return

print(f"სტუდენტი ნომრით {roll\_number} ვერ მოიძებნა.")

def search\_student(self, roll\_number):

# სტუდენტის ძებნა ნომრის მიხედვით

for student in self.students:

if student.roll\_number == roll\_number:

print("სტუდენტის დეტალები:")

print(student.display\_details())

return

print(f"სტუდენტი ნომრით {roll\_number} ვერ მოიძებნა.")

def update\_student\_grade(self, roll\_number, new\_grade):

# სტუდენტის შეფასების განახლება

for student in self.students:

if student.roll\_number == roll\_number:

student.grade = new\_grade

print(f"სტუდენტი ნომრით {roll\_number} შეფასება განახლდა: {new\_grade}")

return

print(f"სტუდენტი ნომრით {roll\_number} ვერ მოიძებნა.")

# მთავარი მენიუ

def main():

sms = StudentManagementSystem()

while True:

print("\nსტუდენტების მართვის სისტემა:")

print("1. ახალი სტუდენტის დამატება")

print("2. ყველა სტუდენტის ნახვა")

print("3. სტუდენტის წაშლა")

print("4. სტუდენტის ძებნა")

print("5. შეფასების განახლება")

print("6. გამოსვლა")

choice = input("აირჩიეთ ოპცია: ")

if choice == "1":

name = input("შეიყვანეთ სტუდენტის სახელი: ")

roll\_number = int(input("შეიყვანეთ სტუდენტის ნომერი: "))

grade = input("შეიყვანეთ სტუდენტის შეფასება: ")

sms.add\_student(name, roll\_number, grade)

elif choice == "2":

sms.display\_all\_students()

elif choice == "3":

roll\_number = int(input("შეიყვანეთ სტუდენტის ნომერი წასაშლელად: "))

sms.remove\_student(roll\_number)

elif choice == "4":

roll\_number = int(input("შეიყვანეთ სტუდენტის ნომერი საძიებლად: "))

sms.search\_student(roll\_number)

elif choice == "5":

roll\_number = int(input("შეიყვანეთ სტუდენტის ნომერი: "))

new\_grade = input("შეიყვანეთ ახალი შეფასება: ")

sms.update\_student\_grade(roll\_number, new\_grade)

elif choice == "6":

print("პროგრამა დასრულდა.")

break

else:

print("არასწორი არჩევანი. გთხოვთ სცადეთ ისევ.")

if \_\_name\_\_ == "\_\_main\_\_":

main()