به نام خدا

سوال دوم

در این سوال شیوه ی دیگری برای برنامه نویسی multi threading در نظر گرفته شده است. یکی از راه های برنامه نویسی multi threading نوشتن کلاسی است که از کلاس Thread ارث بری می کند. در این صورت در کلاس نوشته شده باید تابع run (method) ، override شود. تابع run تابعی است که عملیات هایی که توسط Thread باید انجام شود را مشخص می کند. با شروع شدن Thread یا start این تابع به صورت خودکار صدا زده می شود. در فایل main.py پیوست نمونه ای از این کد را مشاهده می کنید.

در این سوال باید یک آرایه ای از داده های زمان (از ثانیه صفر تا ثانیه ۵۰ با گام های ۰,۰۱) در thread اصلی (main thread) ایجاد شود. دو thread دیگر باید ایجاد شود که این آرایه به عنوان ورودی به آن ها داده می شود. یکی از thread ها cos داده های ورودی و thread دیگر sin داده های ورودی را حساب می کند. بعد از اتمام محاسبات این دو thread، main thread باید حاصل آن ها را با هم جمع کند و حاصل را در یک نمودار نمایش دهد (محور y مقدار cos(t) + sin(t) را نمایش می دهد و محور x زمان یا همان t را).

در این سوال، به طور کلی هدف این هست که بتوان به دو مسیر مجزا کاری مجزا سپرد و در نهایت از خروجی آنها استفاده کرد. زمانی که یک thread بدون اینکه در متد run آن تغییری رخ داده باشد، start می‌شود، تنظیمات پایه متد اجرا میشود. در این سوال، هدف این بود که رفتار متد run به صورتی برنامه‌ریزی شود که در صورت start شدن thread مورد نظر، زمان از ورودی get شود و خروجی در یک حالت کسینوس و در حالتی دیگر سینوس مقدار دریافت شده باشد. برای این کار، ابتدا یک آرایه زمان تعریف شده و به یک صف داده شد تا یکی یکی به threadها سپرده شوند. سپس در thread با ID مختلف تعریف شد که یکی مسئول گرفتن سینوس و یکی کسینوس است. در متد ران، ID خوانده میشود و بر اساس آن عملیات سینوس یا کسینوس برای زمان معین get شده انجام شده و به یک آرایه اضافه میشود. در نهیات این دو آرایه با هم جمع شده و مقدار خروجی آن با استفاده از کتابخانه‌ی matplotlib.pyplot رسم و نمودار خروجی ذخیر میشود.

کد با استفاده از spyder قابل دیباگ نبود. با استفاده از vs-code دیباگ شد و بعد از آن ران شد اما هر بار هیچ خروجی ای نداشت. با استفاده از spyder هم خروجی نمایش داده نمیشود ولی کد به انضمام قرار داده شده است.