



این پروژه، اولین تلاش جدی شما برای مدل کردن یک مسئله با استفاده از مفاهیم طراحی شیء‌گرا است. از این رو توصیه می‌کنیم که پیش از شروع پیاده‌سازی، برای طراحی آن زمان کافی بگذارید.

AP Cup

به زودی قرار است مسابقات جام فوتبال در دورغوزآباد برگزار شود و مسئولین این دوره از ما خواسته‌اند تا پلتفرمی برای پیش‌بینی نتایج پیاده‌سازی کنیم. روند مسابقه به این گونه است که قبل از شروع جام، تیم‌ها در مسابقه ثبت‌نام و اطلاعات بازیکنان و ورزشگاه خود را ثبت می‌کنند. سپس قرعه‌کشی انجام شده و برنامه مسابقات مشخص می‌شود. بازی‌های هر مرحله با قرعه‌کشی مشخص می‌شود. در قالب دو دور رفت و برگشت انجام شده و برندگان به دور بعدی می‌روند و پس از قرعه‌کشی مجدد، مرحله بعدی بازی‌ها انجام می‌شود. این روند تا زمانی که به مسابقه فینال برسیم، ادامه می‌یابد.

توجه: توجه کنید که در این تمرین نیازی به نوشتن تابع main نیست و فایل‌های شما با استفاده از تابع main آماده‌شده توسط دستیاران آموزشی تست خواهد شد؛ برای این منظور شما باید در کلاس اصلی، متدهای به کار رفته در تابع main را پیاده‌سازی کرده باشید. برای مثال به فایل main.cpp که در کنار صورت پروژه قرار گرفته توجه کنید.

کلاس Tournament

برای این که برنامه شما با تابع main مورد نظر ما اجرا شود، لازم است که شما کلاسی به نام Tournament داشته باشید که از آن به عنوان قابلی برای مقدار دهی به برنامه شما استفاده می‌کنیم. دقت کنید که این کلاس صرفاً رابط برنامه شما با تابع main است و به هیچ وجه نباید شامل منطق برنامه شما باشد.

● سازنده (Constructor)

Signature	Tournament::Tournament ();
Example	Tournament ap_cup;

● اضافه کردن تیم (add team)

هر تیم دارای یک نام و یک ورزشگاه است و در هر ورزشگاه تعدادی تماشاچی دارد. هر ورزشگاه نیز دارای نام، ظرفیت، و تأثیرگذاری است.

Signature	void Tournament::add_team(string team_name, string stadium_name, int capacity, float impact, map<string, int> fans_in_stadium);
Example	ap_cup.add_team("Barcelona", "Camp Nou", 100000, 1.0, fans_in_stadium);

map^۱ به زبان ساده، نوعی آرایه است که به جای متناظر کردن یک عدد نامنفی (به عنوان اندیس) به یک مقدار، امکان انتخاب اندیس از هر نوع داده یا ساختار و کلاس را فراهم می‌کند. در اینجا هدف از استفاده از mp، متناظر کردن تعداد تماشاچی‌های یک تیم به ورزشگاه‌های مختلف می‌باشد. در واقع تعداد تماشاگران در هر ورزشگاه در یک map از نام ورزشگاه به تعداد آن‌ها به شما داده می‌شود.

^۱ <http://www.cplusplus.com/reference/map/map/>

● اضافه کردن بازیکنان (add_player)

هر بازیکن دارای اطلاعاتی نظیر نام و قدرت و پست می باشد و به یک تیم اضافه می شود. قدرت عددی بین ۰ تا ۱۰۰ می باشد و بعدها در تعیین قدرت تیم نقش دارد. پست هم یک رشته حاوی یکی از مقادیر striker, defender, midfielder یا goalkeeper می باشد.

Signature	void Tournament::add_player(string team_name, string player_name, int player_power, string player_post);
Example	ap_cup.add_player("Barcelona", "Messi", 93, "striker");

مثال:

دقت کنید که هر تیم برای شرکت در مسابقه ۱۱ بازیکن نیاز دارد و در صورتی که تعداد بازیکنان آن از ۱۱ بیشتر بود، قوی ترین بازیکن ها باید انتخاب شوند؛ به گونه ای که حداقل یک بازیکن در نقش دروازه بان، ۳ بازیکن در نقش مدافع، ۳ بازیکن در نقش هافبک و ۲ بازیکن در نقش مهاجم در تیم حضور داشته باشند (دقت کنید که با توجه به شرایط فوق، تیم باید قوی ترین بازیکنان را در ترکیب داشته باشد) و اگر تعداد بازیکن ها کمتر از ۱۱ نفر بود، تیم اجازه شرکت در مسابقه را نمی یابد.

● شبیه سازی مسابقات (simulate)

کلاس Tournament باید دارای متد simulate باشد و هدف از آن شبیه سازی کل مسابقات و ذخیره اطلاعات آن می باشد.

Signature	void Tournament::simulate();
Example	ap_cup.simulate();

دقت کنید که قبل از اجرای شبیه سازی، در ابتدا باید تیم های حاضر در لیگ را تایید کنیم. برای این کار لازم است که بر روی هر تیم دو شرط چک شود:

۱. آیا تیم دارای ۱۱ بازیکن می باشد؟

۲. آیا تیم در هر پست دارای تعداد بازیکن کافی (که در قسمت قبل به آن اشاره شد) می باشد یا خیر.

اگر تیمی هر یک از دو شرط بالا را نداشت، مجوز حضور در مسابقه را نخواهد داشت و باید پیغامی به شکل زیر در خروجی استاندارد چاپ شود:

Inadequate players in <team_name>

سپس باید بررسی شود که آیا تعداد تیم های واجد شرایط، توانی از دو هست یا نه. اگر این شرط برقرار نبود به چاپ پیغام زیر در خروجی استاندارد، شبیه سازی ادامه نمی یابد.

Inadequate teams

● قرعه کشی

در صورت عدم وجود مشکل در مرحله تأیید، قرعه کشی مرحله اول انجام می شود. برای این منظور تیم ها را به ترتیب لغت نامه ای اسم آن ها مرتب می کنیم و به ترتیب بدست آمده دو به دو جفت می کنیم. به عنوان مثال فرض کنید ۴ تیم a, b, c, d داریم، در این صورت در دور نخست بازی ها تیم های a, b با هم و تیم های c, d نیز با هم بازی می کنند. برای هر جفت تیم، تیمی که در ترتیب لغت نامه ای اول است بازی نخست را میزبان است. قرعه کشی بقیه دورها هم مانند همین دور است که با مرتب کردن تیم های باقی مانده بر اساس ترتیب لغت نامه ای بدست می آید.

● شبیه سازی بازی

برای شبیه سازی نتیجه بازی بین دو تیم a و b، ابتدا قدرت هر تیم را به دست می آوریم. هر تیم سه مشخصه قدرت دارد: defend, midfield و attack که از روی قدرت بازیکنان به دست می آیند. به این صورت که:

$$defend = \lfloor (\frac{goalkeeperPower + \sum defendersPower}{1 + numOfDefenders}) * (\frac{numOfDefenders}{4})^{1/2} \rfloor$$

$$midfield = \lfloor (\frac{\sum midfieldersPower}{numOfMidfielders}) * (\frac{numOfMidfielders}{3})^{1/2} \rfloor$$

$$attack = \lfloor (\frac{\sum attackersPower}{numOfAttackers}) * (\frac{numOfAttackers}{3})^{1/2} \rfloor$$

حضور تماشاگران به هر یک از این سه مشخصه مقداری اضافه می‌کند که از رابطه زیر بدست می‌آید:

$$fansImpact = stadiumImpact * \frac{teamFans}{stadiumCapacity}$$

دقت کنید که اگر جمع تعداد تماشاگران دو تیم از ظرفیت ورزشگاه بیشتر باشد اولویت با تماشاگران تیم میزبان است (تیمی که مالک ورزشگاه است). سپس نتیجه بازی از رابطه زیر محاسبه می‌شود:

$$team_agoals = \lfloor (\frac{attack_a}{defend_b} * \frac{midfield_a}{midfield_b})^{1.5} * 3 \rfloor$$

$$team_bgoals = \lfloor (\frac{attack_b}{defend_a} * \frac{midfield_b}{midfield_a})^{1.5} * 3 \rfloor$$

دقت کنید که هر یک از بخش‌های عبارت فوق باید اعشاری محاسبه شوند. (در صورت نیاز هر تکه از عبارت فوق را به اعداد اعشاری cast کنید)

بازی‌ها به صورت رفت و برگشت خواهند بود. در بازی اول، تیمی که در قرعه‌کشی تیم نخست انتخاب شده بود، میزبان خواهد بود و در بازی برگشت بالعکس. در پایان دو بازی تیمی برنده خواهد بود که در مجموع دو بازی گل بیشتری به ثمر رسانده باشد. در صورت برابری گل‌ها، تیمی برنده خواهد بود که در خانه حریف گل بیشتری زده باشد. در صورت برابری گل‌ها، کار به ضربات پنالتی می‌کشد :) در این حالت دو مقدار زیر را مقایسه می‌کنیم و نتیجه هر کسر که بزرگتر بود، تیم مربوطه برنده بازی می‌باشد.

$$winProbabilityTeam_a = \frac{attack_a}{goalkeeperPowerTeam_b}$$

$$winProbabilityTeam_b = \frac{attack_b}{goalkeeperPowerTeam_a}$$

چاپ نتایج اطلاعات

پس از پایان شبیه‌سازی مسابقات، باید امکان چاپ کردن نتایج وجود داشته باشد. برای این منظور کلاس Tournament باید ۳ متد زیر را داشته باشد:

- چاپ نتایج یک مرحله

Signature	void Tournament::print_round_results(int round_number);
Example	ap_cup.print_round_results(1);

خروجی این دستور باید مانند شکل زیر باشد:

Round <round_number>

<match_number>: // بازی کلیه بازی‌های مرحله

<team_a> <goals_match_1_team_a> - <goals_match_1_team_b> <team_b>

<team_b> <goals_match_2_team_b> - <goals_match_2_team_a> <team_a>

=> winner: <team_name>

- چاپ نتایج کل مسابقات

Signature	void Tournament::print_tournament_results();
Example	ap_cup.print_tournament_results();

خروجی این دستور به ازای کلیه مراحل مسابقه (از اولین مرحله تا آخرین)، باید مانند دستور بالا باشد.

- چاپ نتایج یک تیم

Signature	void Tournament::print_team_results(string team_name);
Example	ap_cup.print_team_results("Barcelona");

این دستور باید نتایج یک تیم را تا مرحله‌ای که پیش رفته چاپ کند. خروجی آن باید مانند دستور بالا باشد ولی با این تفاوت که در هر مرحله تنها اطلاعات مربوط به آن تیم را چاپ کند.

نکات پایانی

- دقت کنید که این کلاس و متدهای این کلاس تنها رابط می‌باشند و برای سهولت شما و درگیر نشدن با خواندن ورودی است و منطق برنامه نباید در آن‌ها پیاده‌سازی شود. در صورت عدم رعایت این نکته، از شما نمره کسر خواهد شد.
- پروژه شما باید حتماً Makefile داشته باشد، و کلاس‌ها در فایل‌های مجزا پیاده‌سازی شوند. در غیر این صورت نمره بخش اجرا را از دست خواهید داد.

نحوه‌ی تحویل

فایل‌های برنامه‌ی خود را با نام A4-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که در آن SID شماره دانشجویی شماست.

لطفاً از روش‌های دیگر فشرده‌سازی مانند rar یا tar.gz استفاده نکنید.

- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم ++g با استاندارد ++c98 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.

- به فرمت و نام فایل‌های خود دقت کنید.

- از صحت فرمت ورودی‌ها و خروجی‌های برنامه‌ی خود مطمئن شوید.

- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.