

## Трећи домаћи задатак из Објектно оријентисаног програмирања 1

1) Написати на језику C++ следећи систем класа. Класе опремити оним конструкторима, деструктором и операторима доделе који су потребни за безбедно и ефикасно коришћење класа. Грешке пријављивати изузецима типа једноставних класа које су опремљене писањем текста поруке. За генеричке збирке није дозвољено коришћење класа из стандардне библиотеке шаблона (STL).

- **Играч** се ствара задатим именом и целобројном вредношћу играча (подразумевано 1000) која може да се дохвати. Могуће је повећати или смањити вредност играча за задати проценат. Могуће је упоредити два играча на једнакост (`igrac1==igrac2`). Играчи су једнаки уколико су им сви атрибути једнаки. Играч се у излазни ток исписује (`it<<igrac`) у облику `име#вредност`.
- **Тим** се ствара задатим називом и максималним бројем играча које је могуће прикључити тиму. При стварању тиму није прикључен ниједан играч. Могуће је прикључити играча на задату целобројну позицију у тиму. Могуће је дохватити тренутни број играча у тиму. Могуће је дохватити (`tim[i]`) играча са задате позиције у тиму. Могуће је дохватити вредност тима као средњу вредност свих прикључених играча. Могуће је упоредити два тима на једнакост (`tim1==tim2`). Два тима су једнака уколико су им називи, максималан број играча и сви прикључени играчи на одговарајућим позицијама једнаки. Тим се у излазни ток исписује (`it<<tim`) у формату `назив [играч1 , играч2 , ... ]`.
- **Привилеговани** тим се ствара са додатно задатом минималном вредношћу играча потребном да се играч прикључи тиму. При прикључивању играча у привилеговани тим, грешка је уколико је вредност играча мања од минималне вредности потребне за прикључење тиму. При исписивању привилегованог тима након назива се додатно исписује минимална вредност играча потребна да се играч прикључи тиму, унутар пара обичних заграда.
- **Пар** је уређен скуп од два показивача на податке неког типа који се задају при стварању и могу појединачно да се дохвате и поставе. Могуће је упоредити два пара истих типова података (`par1==par2`) тако што се упоређују одговарајући показивани подаци. Пар се у излазни ток исписује (`it<<par`) у формату `[податак1-податак2]`.
- **Меч** садржи пар тимова (домаћин и гост) који се задају при стварању и исход меча. Пар тимова је могуће дохватити. Могуће је одиграти меч тако што се на основу вредности тимова поставља исход меча на једну од следећих вредности: `POBEDA_DOMACIN`, `NERESEN`, `POBEDA_GOST`. Уколико постављен исход није нерешен, свим играчима оба тима се ажурирају вредности тако што се играчима победничког тима повећавају, а играчима губитничког тима смањују вредности за 10%. Могуће је проверити да ли је меч одигран. Могуће је дохватити пар целобројних поена које су тимови освојили на основу исхода меча. Пар треба формирати тако да је први податак у пару број поена домаћина, а други податак у пару број поена гостујућег тима. Тим који је победио осваја 3 поена, тим који је изгубио не осваја ниједан поен, док за нерешен исход оба тима освајају по 1 поен. Грешка је уколико меч није одигран. Меч се у излазни ток исписује (`it<<mec`) тако што се исписује пар тимова, а затим исход меча, уколико је меч одигран.

Написати главну функцију која направи неколико играча, дода их у тимове, а потом направи неколико мечева, одигра их и испише их.

---

### НАПОМЕНЕ:

- а) Трећи домаћи задатак је основа за израду треће лабораторијске вежбе.
- б) Студент треба да преда своја решења, сходно упутствима које добије преко мејлинг листе предмета. Предата решења биће доступна студенту и користиће их као полазну тачку за израду лабораторијске вежбе.
- в) Решење домаћег задатка се не оцењује, али улази у састав решења лабораторијске вежбе које се оцењује.