

## Објектно оријентисано пројектовање

### Лабораторијска вежба 2

#### **Задатак 1**

*(Задатак раде групе са терминима у 13.15 и 14.45 часова)*

Нацртати *UML* класни дијаграм за моделовање софтвера који ће помоћи раду ски лифтова на скијалиштима у Србији. За сваки ски лифт памти се ком скијалишту припада, укупан капацитет жице (број прикључака на жици) и назив ски лифта. Сваки ски лифт може имати само један тип прикључака којим се може возити један или више скијаша до врха неке стазе (нпр. двосед, тањир, итд). За сваки ски лифт може се додати нови прикључак, избацити прикључак, идентификовати прикључак са највећим оптерећењем (највећа маса изражена у кг на основу масе скијаша).

Потребно је омогућити чување тренутног стања у бази података. У неким ситуација потребно је извршити ванредни сервис ски лифта којим је могуће променити број прикључака на неком ски лифту (повећати, смањити капацитет ски лифта).

На Старој планини направљена су два ски лифта који као прикључке садрже корпе за четири особе и корпе за две особе. Лифт који садржи корпе за четири особе додатно је осигуран сензором који мери оптерећеност корпе изражену у килограмима. Ски лифт са корпама за двоје мора да памти бројеве ски паса сваког скијаша и унапред се зна максимална маса по корпи која је унапред дефинисана за ту врсту корпе (у овом случају не памти се маса скијаша).

Написати *C++* програм који демонстрира рад пројектоване апликације *UML* класним дијаграмом.

Напомена: Додати потребне методе и атрибуте у класама које су потребне како би апликација функционисала. Обавезно је додавање по неколико атрибута у класама који ближе описују ентитете којима припадају. На *UML* дијаграму обавезно приказати све детаље.

## **Задатак 2**

*(Задатак раде групе са терминима у 16.15, 17.45 и 19.15 часова)*

Нацртати *UML* дијаграм класа који моделује структуру једне рачунарске учионице. Један факултет садржи више рачунарских учионица. У рачунарској учионици се налазе више рачунара.

Свака рачунарска учионица има своју ознаку. Сваки рачунар је упарен са тастатуром и монитором. За сваки монитор памти се дијагонала екрана, резолуција и марка. Сваки рачунар може поседовати бар један процесор, један *RAM* модул и бар један *HDD*.

У циљу повећања перформанси може се убацити и *SSD* који је додатно описан брзином читања и уписа. Рачунарске учионице не морају бити хомогене по питању склопа рачунара.

У рачунарској учионици се налази један 24-портни свич. Мрежним кабловима повезан је сваки рачунар на свич. Потребно је памтити на свичу на ком физичком порту је повезан који рачунар преко своје *MAC* адресе рачунара. Сваки процесор описан је тактом и количином *L1*, *L2* кеша док је *RAM* описан тактом, количином меморије и типом меморије (*DDR3*, *DDR4*, *DDR5*).

Обезбедити да се креира само једна инстанца факултета.

Обезбедити да факултет у сваком тренутку може генерисати извештај о броју рачунара у свакој учионици и извештај о броју рачунара који нису оспособљени за рад по учионицама (за сваки рачунар потребно је памтити да ли је исправан).

Написати *C++* програм који демонстрира рад пројектоване апликације *UML* класним дијаграмом.

Напомена: Додати потребне методе и атрибуте у класама које су потребне како би апликација функционисала. Обавезно је додавање по неколико атрибута у класама који ближе описују ентитете којима припадају. На *UML* дијаграму обавезно приказати све детаље.