

1. Пројектовати на језику *Java* следећи систем класа :

- **Зарубљена купа** се задаје помоћу полупречника доње (r_1) и горње (r_2) основице и висине (h) са подразумеваним вредностима $r_1 = 2$, $r_2 = 1$, $h = 1$. Могу да се дохвате димензије купе, да се израчуна запремина купе ($V = \pi h (r_1^2 + r_2^2 + r_1 r_2) / 3$), да се испита да ли је запремина једне купе мања од запремине друге купе ($k_1 < k_2$), да се димензије купе прочитају из датотеке и да се купа уписује у датотеку у облику $k(r_1, r_2, h)$.
- **Низ** зарубљених купа може да садржи задати број (подразумевано 5) купа. Низ се ствара празан, после чега се купе могу додавати иза последњег попуњеног места, док се низ не напуни (ако се низ препуни, програм се прекида). Може да се дохвати капацитет низа, број попуњених места и може да се испита да ли је низ попуњен до краја. Купе садржане у низу могу да се дохватају и да се проналази индекс купе најмање запремине.

Саставити на језику *Java* главни програм који са главног улаза прочита низ зарубљених купа, проналази купу најмање запремине у низу, испише пронађену купу и њену запремину на главном излазу и понавља претходне кораке док за дужину низа не прочита недозвољену вредност.

2. Написати на језику *Java* следеће класе:

- **Оцена** садржи цео број у опсегу од 5 до 10. Вредности изван опсега приликом стварања промене се у најближу прихватљиву вредност. Може да се дохвати вредност оцене бројчано и словима као и да се оцена испише на главном излазу када се пишу оба облика оцене (на пример: 10 (deset)).
- **Испит** садржи шифру испита од највише 6 знакова и оцену. Може да се дохвати шифра испита и оцена и да се испит испише на главном излазу у облику "шифра:оцена".
- **Студент** има име (текст произвољне дужине), број индекса (дугачак цео број по шеми `ggggrrrr`, где су `g` и `r` цифре године уписа и регистарског броја) и низ испита задатог капацитета (подразумевано 40). Ствара се без иједног испита после чега испити могу да се додају један по један. Повратна вредност при додавању испита показује успех додавања (тј. да ли је било места у низу испита). Не сме да се прави копија студента. Може да се израчуна средња вредност оцена положених испита и да се студент испише на главном излазу у облику "име[годУп/регБр:срОцена]".

Написати на језику *Јава* програм који направи једног студента, додаје му три испита и испише га на главном излазу. Користити само константне податке (не треба ништа учитавати с главног улаза).а

3. Написати на језику *Јава* следеће класе (класе опремити оним конструкторима и оператором за доделу вредности, који су потребни за безбедно коришћење класа; грешке пријављивати изузецима типа једноставних класа које су опремљене писањем текста поруке):

- **Саобраћајница** има име од највише 30 знакова и дужину у метрима. Може да се дохвати име. Може да се упише у излазни ток (`it<<saobracajnica`, пише се име саобраћајнице и дужина у загради). Име и дужина се задају приликом стварања, али име може и накнадно да се мења.
- **Стамбена зграда** има површину основе изражену у m^2 , број спратова и број стамбених јединица по спрату који се задају при стварању зграде. Може да се дохвати површина основе, број спратова, број стамбених јединица по спрату, укупан број стамбених јединица и просечна површина стамбене јединице. Може да се упише у излазни ток (`it<<zgrada`, пише `сеповршина/број_спратова/број_стамбених_јединица`).
- **Улица** је саобраћајница која може да садржи задат број стамбених зграда. Ствара се празна, капацитета одређеног задатом максималном вредношћу целобројног адресног броја, после чега се зграде додају појединачно, под задатим адресним бројем (грешка је ако под задатим адресним бројем већ постоји зграда). Улица не може да се копира. Може да се дохвати укупан број зграда и укупна површина свих стамбених јединица.

Написати на језику *Јава* програм који, читајући податке с главног улаза, направи једно насеље у које дода неколико (2) улица са стамбеним зградама (1-2) и испише насеље на главном излазу.

4. Саставити на језику *Јава* следеће класе:

- Материјална **тачка** у простору се задаје помоћу реалне масе (подразумевано 1) и три реалне координате (подразумевано (0,0,0)). Може да се одреди растојање (r) до друге тачке, да се израчуна привлачна сила између тачке и задате друге тачке ($F=\gamma \cdot m_1 \cdot m_2 / r^2$, $\gamma=6,67 \cdot 10^{-11}$) и да се тачка испише на главном излазу.

- **Низ** материјалних тачака се ствара празан задатог почетног капацитета (подразумевано 5), после чега се тачке додају једна по једна на крај низа. Ако се низ препуни, капацитет му се повећа за 5. Може да се дохвати број тачака у низу, да се дохвати тачка у низу која највише привлачи задату тачку и да се низ испише на главном излазу.

Написати на језику *Java* програм који читајући материјалне тачке с главног улаза направи низ материјалних тачака (читање се завршава уносом негативне масе), испише на главном излазу добијени низ као и тачку која највише привлачи тачку јединичне масе у координатном почетку и понавља претходне кораке све док не прочита празан низ (низ дужине 0).

5. Написати на језику *Java* следеће класе (класе опремити оним конструкторима и оператором за доделу вредности, који су потребни за безбедно коришћење класа; грешке пријављивати изузецима типа класа које су оспособљене за исписивање текста поруке):

- Апстрактно **возило** има задату сопствену тежину. Може да се дохвати назив врсте возила, да се одреди тежина возила и да се возило упише у излазни ток. Пише се назив врсте возила и сопствена тежина возила.
- **Бицикл** је возило.
- **Камион** је возило задате сопствене тежине и носивости. Ствара се без товара после чега може да се дода товар задате тежине и да се скине товар задате тежине. Грешка је ако се камион претовари. Скидањем превише товара тежина товара постане једнака нули. У излазни ток се пише и тренутна тежина товара на камиону.
- Генерички **низ** може да садржи елементе неког типа. Ствара се празан задатог капацитета (подразумевано 20) после чега елементи могу да се додају један по један. Грешка је ако се низ препуни. Може да се дохвати број елемената у низу, да се дохвати елемент задатог редног броја и да се низ испразни. Грешка је ако се покуша дохватити непостојећи елемент.
- Тркачки ауто је возило које садржи генерисан низ вожњи. Ствара се с празним низом капацитета 10 вожњи. Може да се започне нова вожња капацитета 100 етапа, да се последње започетој вожњи дода нова етапа, да се одреди вожња са највећом средњом брзином. У излазни ток се пише и дужина вожње са највећом средњом брзи