

Наследување и виртуелни функции

1. Да се дефинира класа `Celik`, во која ќе се чуваат следниве информации:

- состав (динамички алоцирана низа од знаци),
- тврдина (децимален број),
- производител (низа од 30 знаци).

Во рамките на класата да се дефинираат:

- соодветен конструктор,
- деструктор,
- `get` и `set` функции,
- функција која ќе ја враќа цената на челикот по тон (цената се пресметува како: тврдина \times 1000).

Од оваа класа да се изведе класа `Zapcanik`, за која дополнително ќе се чуваат следниве информации:

- дијаметар во милиметри (децимален број),
- број на запци (целобројна променлива).

Во рамките на класата да се дефинираат:

- конструктор,
- деструктор,
- преоптоварување на операторот `<<` кој ќе го печати растојанието помеѓу два запци на запчаникот,
- да се препокрие функцијата за цената од основната класа која сега ќе ја пресметува цената на запчаникот како: $1000 \times \text{тврдина} \times \text{волумен во } mm^3$ (да се земе дека сите запчаници се со дебелина од 30mm).

Од класата `Celik` да се изведе и класа `Blok_za_motor`, за кој дополнително ќе се чува:

- број на цилиндри (целобројна променлива),
- зафатнина во кубни центиметри (децимален број),
- маса во килограми (децимален број).

Во рамките на класата да се дефинираат:

- конструктор со предефинирани вредности,
- деструктор,
- да се преоптовари операторот `+` кој ќе ја зголемува масата на блокот за децимален број,
- да се препокрие функцијата за цената од основната класа на тој начин што цената на блокот ќе се пресметува како: $20 \times \text{тврдина} \times \text{маса}$.

Да се напише `main` функција со која ќе се тестираат имплементираниите функции во класите.

2. Да се напише класа `Pixel`, која ќе содржи три целобројни промеливи за боја (R, G и B) чии вредности се во опсегот: 0 - 255. Во класата да се дефинираат соодветни конструктори и да се преоптоварат операторот `<<` за печатење на информациите за пикселот.

Дополнително да се напише класа `Matrica`, за која ќе се чуваат динамички алоцирана низа од пиксели и димензии на матрицата (ширина, висина). За оваа класа да се преоптовари операторот `==` за споредба на две матрици според тоа дали имаат еднакви димензии. Да се преоптовари операторот `<<` за печатење на низата од пиксели, како и димензиите на матрицата.

Од класата `Matrica` да се изведе класа `Slika`, класа специјализирана за работа со слики, за која дополнително ќе се чува име на сликата (низа од 30 знаци). Во класата да се дефинираат соодветните конструктори и да се преоптовари операторот `<<` така што ќе се печатат сите информации за сликата. Да се напише функција `fliplr` која ќе врши превртување на сликата лево-десно

Да се напише `main` функција со која ќе се тестираат имплементираниите функции во класите.