Задачи:

Задача 1.

Да се реализира шаблон на клас PriceTag<T>, който представя етикет с цена

на продукт от произволен тип T. За шаблона да се реализират:

● подходящи конструктори, селектори и мутатори;

● операции >> и << за въвеждане на етикет от стандартния вход и извеждане на

стандартния изход;

● член-функция discount, която намалява цената на етикета с даден процент.

Да се реализира шаблон на клас PriceCatalog<T>, който описва ценови каталог на

продукти от тип T, представен с набор от до 200 етикета с цени на продукти. За шаблона да

се реализират:

● подходящ конструктор;

● операция += за добавяне на етикет към каталога;

● операция << за извеждане на информацията в каталога.

Допускаме, че за типа T са реализирани голяма четворка и операции >> и <<.

Задача 2.

Да се реализира клас за пътешествие с полет със самолет AirTravel. Класът съдържа:

- номер на полет flightNumber – низ с произволна дължина от тип char*
- времетраенето на полета duration – задава се в минути

Да се реализира клас за пътешествие с кола CarTravel. Класът съдържа:

- регистрационен номер на превозно средство regNumb – представлява масив от четири цифри;
- брой километри, които ще бъдат изминати distance
- времетраене на пътуването с кола (също с име duration) – задава се в минути

Да се реализира клас за комбинирано пътешествие CombinedTrip, при което има точно едно пътуване с кола и едно пътуване със самолет. Класът притежава всички характеристики на описаните по-горе класове и две допълнителни полета:

- списък от забележителности, които ще бъдат посетени по време на пътуването destinations – представя се чрез динамичен масив от низове от тип char* с произволна дължина
- свободно време за почивка между пътуванията freeTime – задава се в минути

Да се реализира метод getDuration, който връща общото време за пътешествието, което включва времето за пътуване с кола, това за пътуване със самолет и свободното време. Да се реализира метод print, който извежда цялата информацията за пътешествието на екрана.

Задача 3

Реализирайте клас Person, който има име възраст и пол. Добавете негов наследник Student, който има име на университет, факултетен номер, специалност, степен (бакалавър/магистър/докторант) и негов наследник Athlete, който има име на спорт, ранкинг, години опит и булева променлива, пазеща дали играе в професионален отбор. Реализирайте клас StudentAthlete, който наследява студент и атлет и има в себе си име на треньор от университета и брой изиграни мачове. Добавете клас, който съдържа колекция от студенти- атлети и съдържа функция за подбиране на отбор по даден спорт, която принтира топ 5 студентите по ранк в даден университет, които имат поне една година до завършването (приемаме, че бакалавърът е 4 години, магистърът-2, а доктурантурата-3) и не играят в професионален отбор.