

Въведение

За разлика от синтетичната геометрия, която е изградена върху набор от непротиворечащи си аксиоми, аналитичната геометрия изучава геометричните обекти с помощта на т.нар. Декартова координатна система, използвайки принципите на алгебрата и анализа. Аналитичната геометрия е основоположна за различни области на информатиката като например компютърна графика, линейно програмиране и др. Тя се изгражда върху основите на векторната алгебра и на координатния метод. Този метод се прилага за аналитичното представяне на точките, правите и равнините на тримерното евклидово пространство и за изследване на взаимното им разположение.

Постановка

Да се напише програма на C++, която реализира функционалност за следните обекти в пространството: точка (Point), вектор (Vector), и триъгълник (Triangle), чрез класове и наследяване. Да се реализира йерархията: Vector, Triangle -> Point.

Точка

Всяка точка P в тримерното координатно пространство се определя еднозначно от своите координати, които представляват наредена тройка реални числа (x, y, z) . И обратно, всяка наредената тройка от реални числа съответства на единствена точка в пространството при предварително избрана координатна система.

Задача 1. Да се реализира клас Point, който да предоставя средства за създаване на точка. За класа Point да се предефинира оператора ==, който проверява дали две точки съвпадат и ако съвпадат да връща true, а в противен случай false.

Вектор

Всеки свободен вектор в тримерното координатно пространство се представя по единствен начин като линейна комбинация на базисните вектори i, j и k . Коефициентите в това представяне наричаме компоненти на вектора при фиксирана координатна система и представляват наредена тройка реални числа (x, y, z) . Всеки свързан вектор в тримерното координатно пространство се определя еднозначно от две точки - начало A и край B. Ако точката A е с координати (x_A, y_A, z_A) , а B е с координати (x_B, y_B, z_B) , то векторът \vec{AB} е с компоненти $(x_B - x_A, y_B - y_A, z_B - z_A)$.

Задача 2. Да се реализира клас Vector, чиито обект може да се инициализира по два начина – или чрез три реални стойности или чрез две точки. Класът Vector да предоставя член-функции за:

- изчисляване на дължина на вектор, която връща реалното число;
- изчисляване на посока на вектор, която връща единичния вектор или хвърля изключение отпотребителски дефиниран клас за изключения VectorLengthException (наследява стандартния клас за изключения), ако дължината на вектора е нула;

- проверка за нулев вектор, която връща true, ако $x = y = z = 0$ и false в противен случай;
- проверка за успоредност на текущия вектор (x,y,z) и зададен вектор $(v1,v2,v3)$, която връща true, ако $x : v1 = y : v2 = z : v3$ и false в противен случай. Тя хвърля изключение от тип VectorLengthException, ако един от двата вектора е с нулева дължина.
- проверка за перпендикулярност на текущия вектор (x,y,z) и зададен вектор $(v1,v2,v3)$, която връща true, ако $x.v1 + y.v2 + z.v3 = 0$ и false в противен случай. Тя хвърля изключение от тип VectorLengthException, ако един от двата вектора е нулев вектор.

Задача 3. Нека с $a = (x,y,z)$ означим текущия вектор от класа (this). Следните операции с вектори да се реализират като предефиниране на операция, като сами определите дали ще се предефинират като член-функции или като външни функции:

- събиране на два вектора, чрез операция $+$, като резултат се получава отново вектор;
- изваждане на два вектора, чрез операция $-$, като резултат се получава отново вектор;
- умножение на вектор с реално число, чрез операция $*$, като резултат се получава отново вектор;
- скалярно произведение на два вектора, чрез операция $*$, като резултат се получава реално число;
- векторно произведение на два вектора, чрез операция \wedge като резултатът е вектор;
- смесено произведение на три вектора, чрез операция $()$ с аргументи на операцията два вектора и резултат реално число. Така например за пресмятане на $(u \times v) \cdot w$ ще извикваме операцията по следния начин $u(v,w)$;

Триъгълник

Друг геометричен обект е триъгълникът, който се дефинира еднозначно чрез три точки в пространството. При опит да се инициализира триъгълник, чрез точки, някои от които съвпадат, да се изхвърля изключение от тип EqualPointException - потребителски дефиниран клас, който наследява стандартния клас за изключения. EqualPointException съобщава на потребителя, кои са съвпадащите точки, причинили изключението. За класа Triangle да се дефинират член-функции за:

- определяне вида на триъгълника;
- намиране лицето на триъгълника;
- намиране периметъра на триъгълника;
- намиране медицентъра на триъгълника, като връща обект от тип Point;

За класовете Triangle и Point да се предефинират операторите <, > и == по следния начин:

1. ако сравняваме точка и триъгълник с оператор <, то резултатът от операцията ще бъде true, само ако точката лежи в равнината на триъгълника и е от вътрешността му, във всички останали случаи ще връща false;
2. ако сравняваме точка и триъгълник с оператор >, то резултатът от операцията ще бъде true, само ако точката лежи в равнината на триъгълника и е извън него, във всички останали случаи ще връща false;
3. ако сравняваме точка и триъгълник с оператор ==, то резултатът от операцията ще бъде true, само ако точката лежи върху някоя от страните на триъгълника, във всички останали случаи ще връща false.