

ЗАДАЧИ ЗА ЗАДЪЛЖИТЕЛНА
САМОПОДГОТОВКА
ПО
Обектно-ориентирано програмиране
Класове

email: kalin@fmi.uni-sofia.bg

8 март 2017 г.

1. Задача 2.2.31. Да се дефинира клас **BankAccount**, определящ банкова сметка на клиент, състояща се от: име на клиент (символен низ до 50 символа), номер на банкова сметка (символен низ до 20 символа) и налична сума на клиент (реално число). Класът да притежава методи, чрез които може да:

- инициализира банкова сметка;
- извежда на екрана информация за банкова сметка;
- внася пари в банкова сметка;
- тегли пари от банкова сметка.

Да се дефинира оператор за сравнение на две сметки по сумите в тях.

Да се дефинира главна функция, която създава две банкови сметки, извежда информацията в сметките, внася сума в една от сметките и тегли сума от другата сметка.

2. Задача 2.2.39. Да се дефинира клас **Time**, който определя момент от денонощието по зададени час и минути. Класът да съдържа подходящи методи за:

- достъп и промяна на часа и минутите с проверки за коректност;

- добавящ към времето цяло число минути;
- достъп до боря минути, изминали от началото на денонощието;
- оператор за сравнение (казваме, че $t_1 < t_2$, ако t_2 е по-късно в денонощието от t_1).

Да се предефинират операторите $+$, $-$ и $*$, така че да могат да се събират и изваждат две времена, както и да се умножават време с цяло число и цяло число с време. Да се включи дефинираният клас в програма и направят обръщения към член-функциите му и предефинираните оператори.

3. Задача 2.2.44. (асоциативен масив) Да се дефинира клас `Dictionary`, който създава тълковен речник. Тълковният речник се състои от не повече от 500 двойки дума–тълкуване, като думата е символен низ с не повече от 100 символа, а тълкованието е символен низ с не повече от 500 символа.

- Да се дефинира подходяща структура, описваща една двойка дума–тълкуване;
- Да се дефинират подходящи член-данни на клас `Dictionary`;

Клас `Dictionary` да съдържа методи, с които може да се извършват следните операции над речника:

- Инициализация на празен речник;
- извеждане на всички думи в речника и техните тълкувания;
- включване на нова двойка дума–тълкуване в речника;
- изключване на двойка дума–тълкуване от речника (по дадена дума);
- търсене на значението на дадена дума в речник.
- извеждане на всички думи в речника и техните тълкувания по азбучен ред на думите;

Да се дефинира оператор $+$, обединяващ два речника, такъв че:

- Ако някои думи имат значение и в двата речника, значенията да се конкатенират в резултатния сумарен речник;
- Ако общият брой на думите в двата речника надхвърля 500, да се използват само първите 500 думи (при произволна наредба).

4. Да се дефинира структура **Point**, описваща точка в евклидовата равнина и клас **Line**, описващ права в евклидовата равнина, зададена чрез две нейни точки.

Класът **Line** да съдържа методи, чрез които може да се извършват следните операции:

- Проверка дали две прави са успоредни;
- Проверка дали дадена точка лежи на дадена права;
- Намиране на пресечната точка на две прави. Приемаме, че правите не са успоредни. Стойността на резултата може да е произволна в противен случай.
- Създаване на права, която е ъглополовяща на по-големия ъгъл, образуван от две прави. Стойността на резултата може да е произволна в противен случай.

Където е подходящо да се дефинират оператори вместо методи.

Някои от задачите са от сборника *Магдалина Тодорова, Петър Армянов, Калин Николов, "Сборник от задачи по програмиране на C++. Част втора. Обектно-ориентирано програмиране"*. За тези задачи е запазена номерацията в сборника.