

ЗАДАЧИ ЗА ЗАДЪЛЖИТЕЛНА
САМОПОДГОТОВКА
ПО
Обектно-ориентирано програмиране
Класове

email: kalin@fmi.uni-sofia.bg

11 март 2016 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 броя 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, реализиран като масив от двойки дума–тълкуване и методи за:

- създаване на речник;
- извеждане на елементите на речник;
- включване на нова двойка дума–тълкуване в речник;
- изключване на двойка дума–тълкуване от речник;
- търсене на значението на дадена дума в речник.

Да се дефинира оператор $+$, обединяващ два речника. Ако някои думи имат значение и в двата речника, значенията да се конкатенират в резултатния сумарен речник.

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 следните методи:

- Проверка дали две прави са успоредни;
- Проверка дали дадена точка лежи на права;
- Намиране на пресечната точка на две прави, ако има такава;
- Създаване на права, която е ъглополовяща на по-големия ъгъл, образуван от две прави. Приемаме, че правите не са успоредни.

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.