В1 Име _____ф.н. _____ гр. ___

Задача 1.

а)Да се реализира абстрактен базов клас **DataEntry**, който описва обект с данни и дефинира следните операции:

- read: инициализира стойността на обект от стандартния вход;
- print: извежда стойността на обект на стандартния изход;
- bool contains(const char*) const: проверява дали подадената като аргумент стойност се съдържа в началото на данните от даден обект.

б) Да се реализира клас **IntValue**, наследник на **DataEntry**, който дефинира стойност от тип цяло число. Да се дефинира функцията contains така, че ако от низът подаден като аргумент може да се извлече цяло число, функцията да проверява дали то се съдържа в началото на числото от текущия обект.

Например: Ако в текущият обект е записано числото 1234, а се търси низът „12ад“, функцията трябва да върне true. Ако търсеният низ е „23а“, резултатът е false.

в) Да се реализра клас **StringValue**, наследник на **DataEntry**, който дефинира стойност от тип символен низ с произволна дължина. Да се дефинира функцията contains така, че да проверява дали низът записан в текущия обект започва с подадения като аргумент низ.

д) Да се реализира клас **Containter**, който описва множество от разнотипни обекти и дефинира следните операции:

- pushBack: добавя **DataEnrty** към края на контейнер;
- size: намира броя на елементите в даден контейнер;
- []: оператор за индексиране, който дава достъп до елементите на контейнер по индекса на тяхната позиция;
- findEntries(const char*) const: извежда на стандартния изход всички елементи от текущия контейнер, които съдържат в началото на данните си подадения като аргумент низ;
- print: извежда на стандартния изход стойностите на елементите на контейнер.

Контейнерите могат да съдържат произволен брой елементи. Множеството от обекти се представя с динамичен масив.

е) За всички класове, за които се налага, да се реализира канонично представяне.

Задача 2.

Да се реализира клас **ElectricDevice**, който описва електрическо устройство и съхранява данни за името му (символен низ в динамичната памет) и консумирана мощност в киловати (цяло число).

Да се реализира клас **ElectricNet**, който описва електрическа мрежа със зададена максимална консумация на електроенергия. Да се реализират следните оператори:

- + и += с аргумент **ElectricDevice**, който добавя съответното устройство към мрежа, ако тя няма да се претовари, т.е. сумарната консумация на енергията на всички включени в нея устройства няма да надхвърли максималната за мрежата;
- - и -= с аргумент **ElectricDevice**, който премахва съответното устройство от мрежата, ако то е включено в нея. Устройствата се разпознават по името си;
- [] с аргумент символен низ, който дава възможност да се достъпи дадено устройство, включено в мрежата;
- !, който проверява дали има някакво включено в мрежата устройство;
- ++, който удвоява максималната консумация на енергия в мрежата;
- --, който намалява два пъти максималната консумация на енергия в мрежата, ако тя няма да се претовари след тази операция;

За класа **ElectricDevice** могат да се реализират допълнително оператори, съобразно нуждите на долния клас.

В2 Име _____ф.н. _____ гр. ___

Задача 1.

а) Да се реализира абстрактен базов клас **Contact**, който описва потребителски констакт и дефинира следните операции:

- read: прочита информация за потребителски контакт от страндартния вход;
- print: извежда информацията за контакт на стандартния изход;
- bool matchName(const char *name) const: показва дали даден контакт е на потребител с име name.

б) Да се реализира клас **PhoneContact**, наследник на **Contact**, който дефинира телефонен номер на потребител със следните данни:

- name: име на потребителя, символен низ с произволна дължина;
- number: телефонен номер на потребителя, символен низ до 15 символа.

в) Да се реаилизра клас **SkypeContant**, наследник на **Contact**, който дефинира контакт в Skype със следните данни:

- name: име на потребителя, символен низ с произволна дължина;
- skype_name: идентификатор в Skype, символен низ с произволна дължина.

д) Да се реализира клас **ContactBook**, който описва множество от контакти от различен тип и дефинира следните операции:

- push: добавя **Contact** към телефонна книга;
- print: извежда на стандартния всички контакти от телефонна книга;
- findUser (const char *name) const: извежда на стандартния изход всички контакти за потребител с име name.

ContactBook може да съдържа произволен брой елементи.

е) За всички класове, за които се налага, да се реализира канонично представяне.

Задача 2.

Да се реализира клас **Object**, който описва обект и съхранява данни за името му (символен низ в динамичната памет) и неговия обем (реално число).

Да се реализира клас **Container**, който описва съд с определена максимална вместимост (реално число). Да се реализират следните оператори:

- + и += с аргумент **Object**, който добавя обект към съд, ако с това обемът на текущото съдържание няма да надхвърли максималната вместимост на съда;
- - и -= с аргумент **Object**, който премахва обект от съд, ако той е добавен в съда. Обектите се разпознават по техните имена.
- [] с аргумент символен низ, който дава възможност да се достъпи обект, добавен в съда;
- !, който проверява дали има добавен обект в текущия съд;
- ++, който удвоява максималната вместимост на съда;
- --, който намалява два пъти максималната вместимост на съда, ако така обемът на текущото съдържание няма да надхвърли максималната вместимост на съда.

За класа **Object** могат да се реализират допълнително оператори, съобразно нуждите на долния клас.