

Классная работа 1
Классы. Введение.
24.11.2018

Вам необходимо смоделировать телефонную записную книжку. В книжку можно записывать номера, выводить все контакты, совершать звонки по шаблону номера или имени и выводить наиболее популярный контакт.

Класс «контакт»:

```
class Contact{
    string name;           //имя человека
    int prefix;            //префикс номера (число из 3 цифр) 000-333
    int number;            //номер телефона (число из 7 цифр) 0000000-9999999
    int calls;             //количество звонков на этот номер
public:
    Contact();
    void setContact(const string & newName, int newPrefix, int newNumber);
    bool nameLike(const string & partOfName);
    bool numberLike(int numberSuffix);
    int getCalls();
    void newCall();
    void print();
};
```

Необходимо описать:

1. `Contact()` — конструктор по умолчанию (инициализирует все поля);
2. `void setContact(const string & newName, int newPrefix, int newNumber)` — метод для задания (сеттер) имени абонента, префикса и номера (корректность проверять не нужно);
3. `bool nameLike(const string & partOfName);` — метод для проверки совпадения префикса имени абонента с данным параметром, возможно совпадение с целым именем (используйте `string str.substr (from, to)`); например, `A.nameLike("Si")` вернет `true`, если (`A.name == "Sid"`);
4. `bool numberLike(int endOfNumber);` — метод для проверки совпадения суффикса номер абонента с данным параметром, возможно совпадение с целым номером (лучше всего это реализовать в цикле с проверкой всех остатков при делении на 10^K); например, `A.numberLike(567)` вернет `true`, если `A.number == 1234567`;
5. `int getCalls()` — метод для подсчета количества звонков (параметр `calls`);
6. `void newCall()` — метод для отметки о новом звонке (параметр `calls`);
7. `void print()` — метод для печати строки результатов (имя абонента, префикс, номер, количество звонков). Для красивого вывода пользуйтесь манипулятором `std::setw()` из библиотеки `<iomanip>`;

Класс «записная книга»:

```
class PhoneBook{
    int abonentNumber;    //количество записей
    Contact names[10];     //записи
    void printHeader();
public:
    PhoneBook();
    void addAbonent(const string & name, int pref, int number);
```

```

    bool callAbonent(const string & prefixOfName);
    bool callAbonent(int suffixOfNumber);
    void print();
    void printTheMostCommon();
};

```

Необходимо описать:

1. `void printHeader();` — метод для печати строки шапки (имя абонента, префикс, номер, количество звонков). Для красивого вывода пользуйтесь манипулятором `std::setw()` из библиотеки `<iomanip>`;
2. `PhoneBook()` — конструктор (инициализирует поле);
3. `void addAbonent(const string & name, int pref, int number)` — метод для добавления абонента. Если имя абонента содержит более 10 символов, номер содержит более 9 цифр или количество абонентов уже 10, то вывести сообщение о некорректности («FAIL: incorrect name or number format») и ничего не добавлять, иначе — вывести сообщение о корректности («ADD: name phone») добавить запись. Проверять есть ли уже такой абонент в книге НЕ НУЖНО. Вызывается при команде «+ name prefix number».
4. `bool callAbonent(const string & prefixOfName)` — метод для осуществления звонка по началу имени абонента. Вывести всех подходящих (графу `call` оставлять старой). Если подходящих абонентов больше или меньше одного, то сообщить об этом («choose unique» и «no such abonent» соответственно) и не звонить, иначе позвонить (`Contact::newCall()`) и сообщить об этом. Вызывается при команде «#1 name».
5. `bool callAbonent(int suffixOfNumber)` — метод для осуществления звонка по концу номера абонента. Вывести всех подходящих (графу `call` оставлять старой). Если подходящих абонентов больше или меньше одного, то сообщить об этом («FAIL: choose unique» и «FAIL: no such abonent» соответственно) и не звонить, иначе позвонить (`Contact::newCall()`) и сообщить об этом («CALL»). Вызывается при команде «#2 number».
6. `void print()` — метод для печати результатов всех команд с помощью метода `void Team::print()`. Вызывается при команде «?».
7. `void printTheMostCommon()` — метод для поиска и печати абонента, которому звонили больше всего. Если таких абонентов несколько, то печатать всех. Вызывается при команде «*».

Ввод

```
|+ Sid 777 1234567
|+ Scrat 700 1111567
|+ Diego 1000 123
|?
|
|
|#2 4567
|
|
|#2 567
|
|
|#1 Sc
|
|
|#1 Sid
|
|
|*
|
|#1 Diego
|
|.
|
```

Вывод

```
| ADD: Sid 777-1234567
| ADD: Scrat 700-1111567
| FAIL: incorrect name or number format
|name      pref    number  calls
|Sid        777     1234567  0
|Scrat      700     1111567  0
|name      pref    number  calls
|Sid        777     1234567  0
| CALL
|name      pref    number  calls
|Sid        777     1234567  1
|Scrat      700     1111567  0
| FAIL: choose unique
|name      pref    number  calls
|Scrat      700     1111567  0
| CALL
|name      pref    number  calls
|Sid        777     1234567  1
| CALL
|name      pref    number  calls
|Sid        777     1234567  2
|name      pref    number  calls
| FAIL: no such abonent
|
```