

Лабораторная работа 2.

Введение в структуры и классы.

Часть 1.

Выполните следующие задачи по ссылке (максимум 5 баллов):

<https://informatics.msk.ru/mod/statements/view.php?id=91168#1>

Часть 2.

Для каждой задачи заранее подготовьте примеры, аналогичные приведенным в задании.

Задача 1. (1 балл) Реализуйте класс, поддерживающий набор строк в отсортированном порядке. Класс должен содержать два публичных метода:

```
1  class SortedStrings {
2  public:
3      void AddString(const string& s) {
4          // добавить строку s в набор
5      }
6      vector<string> GetSortedStrings() {
7          // получить набор из всех добавленных строк в отсортированном порядке
8      }
9  private:
10     // приватные поля
11 };
12
```

Пример

Код

```
1  void PrintSortedStrings(SortedStrings& strings) {
2      for (const string& s : strings.GetSortedStrings()) {
3          cout << s << " ";
4      }
5      cout << endl;
6  }
7
8  int main() {
9      SortedStrings strings;
10
11     strings.AddString("first");
12     strings.AddString("third");
13     strings.AddString("second");
14     PrintSortedStrings(strings);
15
16     strings.AddString("second");
17     PrintSortedStrings(strings);
18
19     return 0;
20 }
21
```

Вывод

```
1 first second third
2 first second second third
3
```

Задача 2. (1 балл)

Реализуйте класс для человека, поддерживающий историю изменений человеком своих фамилии и имени.

```
1 class Person {
2 public:
3     void ChangeFirstName(int year, const string& first_name) {
4         // добавить факт изменения имени на first_name в год year
5     }
6     void ChangeLastName(int year, const string& last_name) {
7         // добавить факт изменения фамилии на last_name в год year
8     }
9     string GetFullName(int year) {
10        // получить имя и фамилию по состоянию на конец года year
11    }
12 private:
13        // приватные поля
14 };
15
```

Считайте, что в каждый год может произойти не более одного изменения фамилии и не более одного изменения имени. При этом с течением времени могут открываться всё новые факты из прошлого человека, поэтому годá в последовательных вызовах методов `ChangeLastName` и `ChangeFirstName` не обязаны возрастать.

Гарантируется, что все имена и фамилии непусты.

Строка, возвращаемая методом `GetFullName`, должна содержать разделённые одним пробелом имя и фамилию человека по состоянию на конец данного года.

- Если к данному году не случилось ни одного изменения фамилии и имени, верните строку **"Incognito"**.
- Если к данному году случилось изменение фамилии, но не было ни одного изменения имени, верните **"last_name with unknown first name"**.
- Если к данному году случилось изменение имени, но не было ни одного изменения фамилии, верните **"first_name with unknown last name"**.

Пример

Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1965, "Polina");
5     person.ChangeLastName(1967, "Sergeeva");
6     for (int year : {1900, 1965, 1990}) {
7         cout << person.GetFullName(year) << endl;
8     }
9
10    person.ChangeFirstName(1970, "Appolinaria");
11    for (int year : {1969, 1970}) {
12        cout << person.GetFullName(year) << endl;
13    }
14
15    person.ChangeLastName(1968, "Volkova");
16    for (int year : {1969, 1970}) {
17        cout << person.GetFullName(year) << endl;
18    }
19
20    return 0;
21 }
22
```

Вывод

```
1 Incognito
2 Polina with unknown last name
3 Polina Sergeeva
4 Polina Sergeeva
5 Appolinaria Sergeeva
6 Polina Volkova
7 Appolinaria Volkova
8
```

Задача 3. (2 балла)

Дополните класс из предыдущей задачи «Имена и фамилии — 1» методом `GetFullNameWithHistory`:

```
1 class Person {
2     public:
3     void ChangeFirstName(int year, const string& first_name) {
4         // добавить факт изменения имени на first_name в год year
5     }
6     void ChangeLastName(int year, const string& last_name) {
7         // добавить факт изменения фамилии на last_name в год year
8     }
9     string GetFullName(int year) {
10        // получить имя и фамилию по состоянию на конец года year
11    }
12    string GetFullNameWithHistory(int year) {
13        // получить все имена и фамилии по состоянию на конец года year
14    }
15 private:
16    // приватные поля
17 };
18
```

В отличие от метода `GetFullName`, метод `GetFullNameWithHistory` должен вернуть не только последние имя и фамилию к концу данного года, но ещё и все предыдущие имена, и фамилии в обратном хронологическом порядке. Если текущие факты говорят о том, что человек два раза подряд изменил фамилию или имя на одно и то же, второе изменение при формировании истории нужно игнорировать.

Для лучшего понимания формата см. примеры.

Пример 1

Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1900, "Eugene");
5     person.ChangeLastName(1900, "Sokolov");
6     person.ChangeLastName(1910, "Sokolov");
7     person.ChangeFirstName(1920, "Evgeny");
8     person.ChangeLastName(1930, "Sokolov");
9     cout << person.GetFullNameWithHistory(1940) << endl;
10
11     return 0;
12 }
13
```

Вывод

```
1 Evgeny (Eugene) Sokolov
2
```

Пример 2

Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1965, "Polina");
5     person.ChangeLastName(1967, "Sergeeva");
6     for (int year : {1900, 1965, 1990}) {
7         cout << person.GetFullNameWithHistory(year) << endl;
8     }
9
10    person.ChangeFirstName(1970, "Appolinaria");
11    for (int year : {1969, 1970}) {
12        cout << person.GetFullNameWithHistory(year) << endl;
13    }
14
15    person.ChangeLastName(1968, "Volkova");
16    for (int year : {1969, 1970}) {
17        cout << person.GetFullNameWithHistory(year) << endl;
18    }
19
20    person.ChangeFirstName(1990, "Polina");
21    person.ChangeLastName(1990, "Volkova-Sergeeva");
22    cout << person.GetFullNameWithHistory(1990) << endl;
23
24    person.ChangeFirstName(1966, "Pauline");
25    cout << person.GetFullNameWithHistory(1966) << endl;
26
27    person.ChangeLastName(1960, "Sergeeva");
28    for (int year : {1960, 1967}) {
29        cout << person.GetFullNameWithHistory(year) << endl;
30    }
31
32    person.ChangeLastName(1961, "Ivanova");
33    cout << person.GetFullNameWithHistory(1967) << endl;
34
35    return 0;
36 }
37
```

Вывод

```
1 Incognito
2 Polina with unknown last name
3 Polina Sergeeva
4 Polina Sergeeva
5 Appolinaria (Polina) Sergeeva
6 Polina Volkova (Sergeeva)
7 Appolinaria (Polina) Volkova (Sergeeva)
8 Polina (Appolinaria, Polina) Volkova-Sergeeva (Volkova, Sergeeva)
9 Pauline (Polina) with unknown last name
10 Sergeeva with unknown first name
11 Pauline (Polina) Sergeeva
12 Pauline (Polina) Sergeeva (Ivanova, Sergeeva)
13
```

Пример 3

Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1965, "Polina");
5     person.ChangeFirstName(1965, "Appolinaria");
6
7     person.ChangeLastName(1965, "Sergeeva");
8     person.ChangeLastName(1965, "Volkova");
9     person.ChangeLastName(1965, "Volkova-Sergeeva");
10
11     for (int year : {1964, 1965, 1966}) {
12         cout << person.GetFullNameWithHistory(year) << endl;
13     }
14
15     return 0;
16 }
```

Вывод

```
1 Incognito
2 Appolinaria Volkova-Sergeeva
3 Appolinaria Volkova-Sergeeva
4
```

Задача 4. (2 балла)

Реализуйте класс `ReversibleString`, хранящий строку и поддерживающий методы `Reverse` для переворота строки и `ToString` для получения строки.

Пример

Код

```
1 int main() {
2     ReversibleString s("live");
3     s.Reverse();
4     cout << s.ToString() << endl;
5
6     s.Reverse();
7     const ReversibleString& s_ref = s;
8     string tmp = s_ref.ToString();
9     cout << tmp << endl;
10
11     ReversibleString empty;
12     cout << "" << empty.ToString() << "" << endl;
13
14     return 0;
15 }
16
```

Вывод

```
1 evil
2 live
3 ""
4
```

Задача 5. (2 балла)

Дополните класс `Person` из задачи 2 конструктором, позволяющим задать имя и фамилию человека при рождении, а также сам год рождения. Класс не должен иметь конструктора по умолчанию.

При получении на вход года, который меньше года рождения:

- методы `GetFullName` и `GetFullNameWithHistory` должны отдавать **"No person"**;
 - методы `ChangeFirstName` и `ChangeLastName` должны игнорировать запрос.
- Кроме того, необходимо объявить константными все методы, которые по сути ими являются.

Пример

Код

```
1  int main() {
2      Person person("Polina", "Sergeeva", 1960);
3      for (int year : {1959, 1960}) {
4          cout << person.GetFullNameWithHistory(year) << endl;
5      }
6
7      person.ChangeFirstName(1965, "Appolinaria");
8      person.ChangeLastName(1967, "Ivanova");
9      for (int year : {1965, 1967}) {
10         cout << person.GetFullNameWithHistory(year) << endl;
11     }
12
13     return 0;
14 }
15
```

Вывод

```
1  No person
2  Polina Sergeeva
3  Appolinaria (Polina) Sergeeva
4  Appolinaria (Polina) Ivanova (Sergeeva)
5
```