

ВВЕДЕНИЕ

Объектно-ориентированное программирование — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования. Язык C++, используемый в ходе данного проекта, позволяет отлично реализовать все основные парадигмы объектно-ориентированного программирования.

C++ — компилируемый, строго типизируемый язык программирования общего назначения. Поддерживает разные парадигмы программирования: процедурную, обобщённую, функциональную. Наибольшее внимание уделено поддержке объектно-ориентированного программирования. Компиляторы C++ есть на каждой операционной системе, большинство программ легко переносится с платформы на платформу, со средой разработки и библиотеками у вас точно не возникнет проблем. C++ отлично поддерживается, и речь не только об обновлениях стандартов и постоянном внедрении улучшений. Язык имеет колоссальное сообщество программистов, которые постоянно делятся библиотеками, шаблонами и кодами, приходят на помощь новичкам и опытным коллегам. Также широко известен тот факт, что C++ программисты широко востребованы по всему миру!

Основными концепциями объектно-ориентированного программирования являются: абстракция, инкапсуляция, полиморфизм и наследование. Язык C++ позволяет реализовать их все в полной мере.

Итак, C++ отличается высокой производительностью, кроссплатформенностью, многопоточностью, доступностью литературы, бесплатностью и универсальностью.

Стоит отметить, что C++ — чрезвычайно мощный язык, содержащий средства создания эффективных программ практически любого назначения, от низкоуровневых утилит и драйверов до сложных программных комплексов самого различного назначения.

К недостаткам языка C++ можно отнести сложность и избыточность, из-за которых язык трудно изучать, а построение компилятора сопряжено с большим количеством проблем.

Тем не менее, язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы.

1.ОБЗОР ЛИТЕРАТУРЫ

1.1 Постановка задачи

Программа должна иметь удобный пользовательский интерфейс, с включённым меню. Вся информация должна находиться в различных файлах, связанных определённым образом. Требуется реализовать функции создания культурных заведений, их кухонь, рейтингов, пунктов меню, добавления рейтинга культурному заведению, сортировки культурных заведений по рейтингу, редактирования культурных заведений, удаления культурных заведений, добавления позиций в меню, вывода всех имеющихся культурных заведений на экран, записи всех имеющихся культурных заведений в файл, чтения культурных заведений из файла.

В программе будет использоваться иерархия классов с использованием наследования. Будут разработаны классы контейнеров, итераторов и алгоритмов, будет проведена обработка исключений.

Путеводитель по культурным заведениям может показаться ненужным, но на самом деле трудность в выборе правильного культурного заведения может вызывать огромный стресс у большого количества людей. Плохо выбранное место для проведения вечера может испортить настроение, а иногда даже отношения между людьми. Однако, использование интерфейса путеводителя по культурным заведениям поможет вам избежать дорогостоящих ошибок.

Программа-путеводитель по культурным заведениям поможет вам правильно выбирать места для культурного отдыха, имея огромный выбор различных мест для времяпрепровождения.

Для реализации программы используется объектно-ориентированный язык C++, среда разработки JetBrains CLion. Приложение написано для ОС Windows 10.

1.2 Обзор предметной области

Благодаря разработке данного проекта мы сможем углубить свои знания по изученным дисциплинам, получим незабываемый опыт создания уже готового продукта.

Данный проект можно реализовать на множестве языков программирования, таких как Java, Assembler, Swift, но лучше всего для этой цели подходит язык C++. Именно он позволяет лучше всего реализовать парадигмы объектно-ориентированного программирования, которые так востребованы в данном курсовом проекте.

Также немаловажную роль играет тот факт, что C++ распространён повсеместно. Многие невероятно популярные платформы были разработаны именно на это языке, к примеру Google Chrome или Adobe Flash Player.

Ещё одним из факторов, благодаря которому C++ является фактически идеальным решением для выполнения данного курсового проекта является предсказуемое выполнение программ. Весь код, неявно генерируемый компилятором для реализации языковых возможностей определён в стандарте. Также строго определены места программы, в которых этот код выполняется. Это даёт возможность замерять или рассчитывать время реакции программы на внешнее событие.

Язык C++ также отличается возможностью работы на низком уровне с памятью, адресами, портами, ссылками. При неосторожной работе это может легко превратиться в недостаток языка, но при правильном использовании может стать огромным плюсом, так как с помощью этого можно реализовывать множество сложнейших процедур с помощью использования простейших операций.

На основе всего вышесказанного, можно постановить что язык C++, несмотря не все его немалочисленные недостатки, действительно является отличным средством для реализации данного курсового проекта.

2.СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

2.1 Блок меню приложения

Этот функциональный блок отвечает за взаимодействие пользователя с приложением. Он выводит на экран меню с немалым выбором операций. Среди выводящихся операций можно найти: операцию добавления культурного заведения, операцию добавления рейтинга культурному заведению, операцию вывода культурных заведений на экран, операцию записи имеющихся культурных заведений в файл, операцию чтения культурных заведений из файла, операцию редактирования культурного заведения, операцию удаления культурного заведения, операцию сортировки культурных заведений по рейтингу и тому подобное. Блок имеет защиту от некорректного ввода, и будет продолжать бездействовать вплоть до введения правильных данных. Данный блок связан с блоком взаимодействия с файловой системой путём считывания информации о культурных заведениях и их дополнительных параметрах из файла `cultural_places.txt`.

2.2 Блок файловой системы

Блок ответственен за чтение информации из файла, и запись информации в файл. В файле изначально приведён список рекомендуемых культурных заведений с экспертным рейтингом, но каждый пользователь имеет право добавлять свои заведения, изменять их рейтинг как хочет, и создавать свои файлы со своими подборками культурных заведений. Благодаря этому блоку пользователь может сохранять введённые данные, чтобы впоследствии не записывать их в программу вновь. Он напрямую связан с блоком меню, ведь через блок меню проходит любая информация, добавляемая в список культурных заведений, который после записывается в файл.

Структурная схема представлена в приложении А.

3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

3.1 Структура данных файла cultural_places.txt

Таблица 3.1 – структура данных файла cultural_places.txt

name	cuisine	menu	rate
string	string	menu	double

В данной таблице хранится информация о структуре данных файла cultural_places.txt, который хранит предложенные экспертом культурные заведения.

3.2 Описание класса cultural_place

Класс cultural_place можно назвать основным классом данного проекта, ведь проект посвящён именно культурным заведениям. Рассмотрим поля и методы данного класса:

- поле string name: поле которое хранит название культурного заведения. Впоследствии название заведения используется для реализации функции поиска по имени, и для создания проверки, чтобы не было создано два заведения с одинаковым названием.

- поле string cuisine: поле в котором хранится кухня заведения. Данное поле необходимо чтобы предоставить больше информации пользователю, ведь кухни разных стран довольно сильно отличаются.

- поле menu menu: нетрудно догадаться, что в данном поле описывается меню заведения. Думаю, излишне расписывать причины его необходимости.

- метод `add_rate`: вызывает функцию, которая принимает в себя дополнительную оценку рейтинга, и вычисляет средний рейтинг данного заведения.

- метод `get_rate`: вызывает функцию, которая получает параметр `rate` у данного заведения.

- метод `sort_cultural_places_by_rate`: этот метод нужен, чтобы отсортировать культурные заведения по рейтингу. В данном проекте используется метод сортировки двусвязного списка пузырьком.

-метод `find_by_name`: метод, который используется во многих частях программы, принимает в себя строку (имя заведения, которое нужно найти), и ищет по имени данное заведение среди двусвязного списка культурных заведений. Используется в проверке, на то чтобы не создавалось 2 заведения с одинаковым именем, в функции удаления заведений, и в функции редактирования заведения.

3.3 Описание класса `program`

В этом классе нету ни одного поля, но его методы как никакие другие важны для курсового проекта, поэтому я и решил его описать.

-конструктор класса `program`: в нём выделяется память под двусвязный список культурных заведений, впоследствии память очищается в деструкторе.

-метод `run`: метод, который, можно сказать, приводит программу в действие. Именно после его выполнения на экран выводится меню, в котором пользователь и выбирает, чего ему угодно сделать с культурными заведениями.

-приватный метод `init_program_menu`: гвоздь программы. Вызывает создание всех команд, что должны быть в меню, в том числе команды добавления культурного заведения, добавления рейтинга культурного заведения, вывода на экран всех имеющихся культурных заведений, записи

всех имеющихся культурных заведений в файл, чтения всех культурных заведений из файла, редактирования существующего культурного заведения, удаления существующего культурного заведения, сортировки всех имеющихся культурных заведений по их рейтингу, и, разумеется, функцию выхода из программы.

3.4 Описание класса `linked_list`

Данный класс является шаблонным, написанным вручную классом-контейнером, в котором мы храним наши культурные заведения. Класс был реализован в соответствии со всеми необходимыми для работы программы требованиями. В нём используются многие поля и методы, я считаю нужным описать здесь только основные из них. В списке также реализован итератор, для быстрого и качественного прохода по нему, что используется в некоторых методах. Перегружены операторы `++`, `==`, `!=`, `bool`, `*`, `+`.

- класс `Node`: данный класс позволяет реализовать структуру двунаправленного списка. Смысл двунаправленного списка состоит в том, что каждый элемент имеет указатель на следующий элемент, и указатель на предыдущий. Также в списке имеется указатель на начало, и на конец списка, так называемые `head` и `tail`.

- метод `init`: необходим для инициализации списка, в нём “голове” и “хвосту” списка присваивается `nullptr`, а размер списка устанавливается равным нулю. Данный метод вызывается в конструкторе класса.

-метод `push_back`: благодаря этому методу мы можем увеличивать вместимость нашего списка. Он выделяет память для ещё одного элемента списка, и записывает в список необходимый элемент.

-метод `remove`: этот метод нам понадобится если мы захотим удалить некий элемент списка. В нём проверяется, не является ли элемент, который нужно удалить, началом или концом списка, и если он действительно не является ни тем ни другим, то в методе происходит переадресация указателя предыдущего элемента “`next`” на следующий элемент (после того, который мы хотим удалить). Аналогично, указатель следующего элемента `previous`

начинает указывать на предыдущий элемент, таким образом мы и удаляем необходимый элемент из списка.

- метод `sort`: этот метод представляет собой реализацию сортировки культурных заведений по рейтингу. Сортировка реализована пузырьковым методом.

- метод `get_size`: вам может показаться что этот метод возвращает размер двусвязного списка. И тогда вы будете абсолютно правы.

4. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

4.1 Описание алгоритмов по шагам

4.1.1 Алгоритм описания по шагам функции `delete_cult_zav()`

1. Начало.
2. Вызывается метод функции `execute` .
3. Передаём в метод `execute` указатель на двусвязный список культурных заведений.
4. Создаём переменную `name` для хранения имени заведения, которое мы хотим удалить.
5. Требуем ввод имени культурного заведения для удаления.
6. Создаём переменную `a`, которой присваиваем тип ссылки на объект.
7. Вызываем функцию `find_by_name(name)`.
8. В функции `find_by_name(name)` проходимся по двусвязному списку с помощью итератора, находим элемент с таким же параметром `name`, и возвращаем ссылку на него.
9. В случае ненахождения элемента с таким же параметром `name` мы выводим на экран ошибку ввода.
10. Присваиваем переменной `a` ссылку, полученную из функции `find_by_name(name)`.
11. Вызываем функцию `delete_cultural_place(&a)`.
12. В функции `delete_cultural_place(cultural_place * cultural_place)` вызываем функцию `remove(cultural_place)` у объекта двусвязного списка.
13. В функции `remove(T *item)` мы проходимся по двусвязному списку, пока не найдём тот элемент, который мы хотим удалить.
14. Как только мы находим этот объект, мы перемещаем указатель `next` предыдущего объекта на следующий объект, и потом перемещаем указатель `previous` следующего объекта на предыдущий объект.
15. Конец.

5.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

В проекте было реализовано консольное приложение для Windows.

5.1 Минимальные требования к программному и аппаратному обеспечению

Для использования приложения необходим компьютер со следующими минимальными характеристиками:

- ОС Windows 7 и более поздние версии.
- ОЗУ 512 МБ и более.
- Разрешение экрана не меньше чем 1028x768

5.2 Руководство по использованию

Сразу после запуска программы, у пользователя на мониторе появляется меню, с избыточным количеством команд. Пользователь может воспользоваться любой из них, с разной степенью успешности:

```
1 - Add cultural place
2 - Add rate to a cultural place
3 - Show cultural places
4 - Write to file
5 - Read from file
6 - Update cultural place
7 - Delete cultural place
8 - Sort cultural places by rate
9 - Exit
```

Пользователю достаточно ввести цифру, которая находится напротив необходимой ему команды. При неправильном вводе программа выведет на экран ошибку и попросит пользователя попытаться ввести снова:

```
1 - Add cultural place
2 - Add rate to a cultural place
3 - Show cultural places
4 - Write to file
5 - Read from file
6 - Update cultural place
7 - Delete cultural place
8 - Sort cultural places by rate
9 - Exit
Kazakhstan
Error input :(
Try again:|
```

После ввода цифры 1, и нажатия на клавишу enter вы можете добавить в список культурное заведение. Для этого вам нужно будет ввести его имя, кухню, и количество позиций в меню. Вы можете добавлять культурные заведения сколько душе угодно, но при этом будет запрещено создавать заведения с одинаковым именем.

Нажав на цифру 2, вы сможете добавить рейтинговую оценку культурному заведению. Для этого вам понадобится ввести его имя. При вводе несуществующего заведения будет выведена ошибка, и будет совершён возврат в главное меню.

```
1 - Add cultural place
2 - Add rate to a cultural place
3 - Show cultural places
4 - Write to file
5 - Read from file
6 - Update cultural place
7 - Delete cultural place
8 - Sort cultural places by rate
9 - Exit
2
Enter the name of cultural place:Spichki
Enter rate:10
```

После нажатия на клавишу 3, на экран красиво выведутся все имеющиеся культурные заведения.

3

Name: Spichki

Cuisine: USA

Rate: 10

Name: Craftman

Cuisine: England

Rate: 0

Name: Pinta

Cuisine: French

Rate: 0

Name: Clever

Cuisine: Irish

Rate: 0

Нажатие на 4ую клавишу даст пользователю возможность записать все имеющиеся заведения в текстовый файл.



Клавиша 5 даёт возможность считать нужную информацию из файла. Для этого потребуется ввести название файла, из которого вы хотите считать данные. При введении названия несуществующего файла, команда работать откажется.

При нажатии на бую клавишу вы можете редактировать уже имеющееся культурное заведение. Для этого вам нужно вписать имя заведения, которое вы хотите изменить, и заново ввести всю информацию о нём. При введении некорректного имени, на экран выведется ошибка, и программа автоматически перейдёт в главное меню.

7ая клавиша даёт возможность удалить выбранное культурное заведение. Для этого вам понадобится всего лишь ввести его имя. Если будет введено некорректное имя, функция, очевидно, работать не будет.

8ая клавиша может предложить вам сортировку по рейтингу всех имеющихся культурных заведений.

```
-----  
Name: Craftman  
Cuisine: England  
Rate: 10  
-----  
-----  
Name: Spichki  
Cuisine: USA  
Rate: 9  
-----  
-----  
Name: Pinta  
Cuisine: French  
Rate: 8  
-----  
-----  
Name: Clever  
Cuisine: Irish  
Rate: 7  
-----
```

9ая клавиша обеспечит вам выход из программы.

9

It was a pleasure to deal with you

Process finished with exit code 0

6. СПИСОК ЛИТЕРАТУРЫ

1. Шилдт Г. Самоучитель C++: Пер. с англ. — 3-е изд. — СПб.: БХВ-Петербург, 2003. — 688 с.

2. Харви Дейтел, Пол Дейтел Как программировать на C++/2017г.

3. Страуструп, Б. Язык программирования C++/ Б.Страуструп; специальное издание. Пер. с англ. — СПб.: ВHV, 2008. — 1098 с.

ЗАКЛЮЧЕНИЕ

При использовании приложения-путеводителя по культурным заведениям вы можете получить гору преимуществ, таких как:

1. Удобное составление списков подходящих культурных заведений.
2. Пользование системой объективных оценок, оставленных пользователями.
3. Запись необходимой информации о культурных заведениях в файл в удобной форме, для хранения необходимой информации.

Очевидно, идеала добиться невозможно, поэтому расскажу как можно улучшить данное приложение:

1. Добавить систему аккаунтов пользователей и администраторов приложения.
2. Реализовать сортировку по алкогольным напиткам.
3. Подключить графический интерфейс, для лучшего понимания приложения пользователем.

Я и сам рад бы реализовать все эти функции, но я большую часть времени посвятил сбору теоритических данных на практической основе.

Приложение было написано в среде JetBrains CLion , так как я считаю что именно эта среда разработки лучше всего подходит для реализации программ на языке C++.

Системные требования:

- Операционная система Windows 7 и выше;
- Процессор Intel Pentium III 1 GHz;
- Оперативная память 500 MB;
- Свободное место на жестком диске:5 MB