**1. Постановка задачи**

Запрограммируйте шаблонный класс, реализующий стек. Класс должен поддерживать следующие операции:

1. Помещение объекта в стек;

2. Извлечение объекта из стека;

3. Получение размерности стека.

В случае попытки вызова операции извлечение объекта из стека при условии, что стек пуст,

должно генерироваться исключение класса EStackEmpty(наследник класса EStackException ,

пример приведен в документе ForTask\_1 ). Данный класс должен содержать публичный метод char\* what(),

возвращающий диагностическое сообщение.

**2. Предлагаемое решение.**

Предлагаемое решение излагается на уровне идеи. Описывается общее решение задачи, с

указанием структуры решения, в частности описание предлагаемых сущностей(классы),

функций.

Если в качестве решения предлагается архитектура ПО, она также описывается и

представляется с помощью UML диаграммы. Если предложенная архитектура ПО

базируется на одном их известных паттернов, то в отчете обязательно описывается

назначение этого паттерна его архитектура, а также какие шаги были выполнены для

применения этого паттерна для решения задачи. (Таким образом сдаете теоретическую

часть, рассматриваемой темы + ваше решение)

a) Реализация шаблонного класса, это позволит задавать тип для объектов.

Для создания класса стек я использовал однонаправленный список, узел которого состоит из двух полей,

указатель на пердыдущий узел и сам ключ.

Стек - это область памяти , в которой функция хранит аргументы и адрес возраста.

В стеке существуют такие методы:

void Print() - это метод перебора значений,я использую его для вывода стека в консоль

аргументом для данной функции является аргумент функции обратной связи.

voit Push() - это метод помещения объекта в стек,получаем значение которое будем добавлять в стек,

создаем пустое звено. Новое звено создается со значением для которого предыдущий узел, это последний элемент стека

и после делаем его последним. Делаем счетчик.

Pop() - метод извлечения объекта из стека , метод удаляет узел и из-за этого возвращает не ссылкой, а константой.

Делаем счетчик.

**3. Коды программ. полностью прокомментированы, с включением объяснений решения**

**(фактически это ваш устный ответ).**

Были использованны следующие методы с комментариями:

estackexception.h

estackempty.h

stack.h

**4. Инструкция пользователя. В этом разделе демонстрируются варианты взаимодействия**

**с вашим ПО с точки зрения пользователя. Если требуется, то предоставляете готовые**

**примеры (входные файлы с исходными данными и т.д.).**

С точки зрения пользователя , он может добавить элемент, удалить элемент или вывести.

**5. Тестирование. В этом разделе описывается процесс тестирования, в частности все**

**случаи (case1, case2) допустимые и недопустимые при которых ваше ПО выдает**

**адекватную реакцию. Также, тестовые примеры, относительно которых проводилось**

**тестирование.**

Я проверил вызов исключения, создал объект класса и попытался вывести элемент из стека,

программа вызвала исключение.

Stack<string>st;

St.Pop();

Так как стек пустой, я решил проверить правильность работы вызова исключений, для этого я вызвал метод Pop(),

итог, исключение не вызвалось, программа работает корректно.

Stack<string>st;

st.Push("123456");

st.Push("abcd");

st.Push("xyz");

St.Pop();

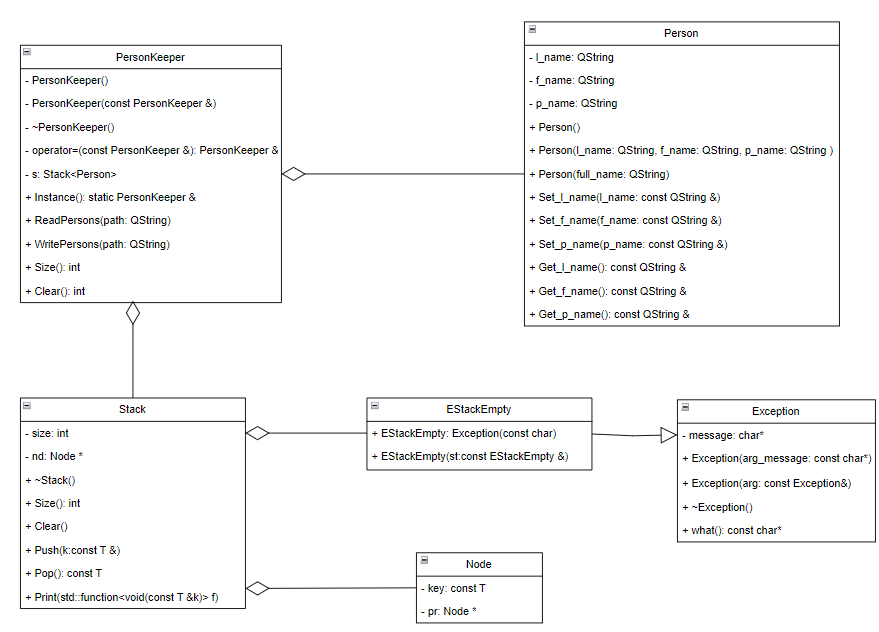
**2 часть**

1. Реализация класса PersonKeeper,Он реализован в соответсвии с шаблоном Singleton. В нем есть такой метод как readPersons(считывает данные из входного потока

и создает объекты класса Person c дальнейшим размещением их в стек).Входной файл представляет из себя Фамилия Имя Отчество,

в качестве разделения выступают пробелы.

А так же есть метод writePerson, возвращает стек,он записывает в поток из стека данные о людях

2. UML- диаграмма

Смысл паттерна Singleton заключается в том , что класс состоит в одном экземпляре и его методы находятся

в приватной части.

Метод instance использую для получения доступа к экземпляру класса.

Нужно реализовать класс Person, он хранит в себе фио, я использую конструктор,

который получает фио в виде строки и разбивает строку на фамилию имя и отчество.

Для этого я использую функцию split(QRegularExpression("\\s+"), Qt::SkipEmptyParts);

Так же проверяем корректное ли получилось разбиение.

Как сказанно выше , я использовал для класса PersonKeeper два метода(чтение и запись).

Методу чтения на вход поступает путь, в котором находится файл, для считывания из файла использую readLineInfo.

Эта функция считывает имена построчно.

Методу записи так же задаем путь, поскольку данные хранятся в стеке, делаю перебор значения методом Print() с дальнейшей записью через пробел.

3.

Использовал файлы:

Person.h

Person\_keeper.h

4.

Пользователь изначально может считать данные из файла в стек, потом он может записать данные из стека в файл,

еще он может просмотреть количество имен.

5.

изначально я создал файл с полным фио, программа сработала корректно.

Morozov Pavel Olegovich

Bakuto Liliya Maksimovna

Evdokimov Anton Alekseevich

потом исключил отчество, ошибок не выявленно

Morozov Pavel

Bakuto Liliya

Evdokimov Anton