

Лабораторная работа №1

«Основы объектно-ориентированного программирования»

Цель работы: изучить основы объектно-ориентированного программирования (понятия инкапсуляции, областей видимости, членов класса, наследования, полиморфизма, событийного программирования, перегрузки функции и виртуальных методов), а также ознакомиться с библиотеками для создания оконных приложений в QT C++.

Программа должна обеспечивать ввод/вывод данных с помощью визуальных компонентов, контроль ошибок при вводе, обработку исключительных ситуаций.

Задание 1.

Индивидуальные задания

Описать класс-родитель и класс-потомок, имеющие методы, указанные в соответствующем варианте задания (потомок наследует или переопределяет методы родителя и приобретает новые). Предусмотреть необходимое количество кнопок для демонстрации каждого из методов объектов.

Данные задания выполнять в соответствии с выданным вариантом:

1. Нарисовать вращающееся колесо. Родительский класс – перемещающийся круг.
2. Нарисовать повозку (прямоугольник на 2 колесах). Родительский класс – перемещающийся прямоугольник.
3. Нарисовать ракету с пламенем из сопла. Родительский класс – перемещающийся отрезок.
4. Нарисовать лицо двигающую глазами и открывающую рот. Родительский класс – перемещающийся эллипс.
5. Нарисовать солдата, перемещающегося и отдающего честь. Родительский класс – перемещающийся прямоугольник.
6. Нарисовать корабль, который может поднимать флаг. Родительский класс – перемещающийся прямоугольник.
7. Нарисовать автомобиль с открывающимися дверями и включающимися фарами. Родительский класс – перемещающийся прямоугольник.
8. Нарисовать сигнальщика, подающего различные сигналы. Родительский класс – перемещающийся прямоугольник.
9. Нарисовать самосвал, который может поднимать кузов. Родительский класс – перемещающийся прямоугольник.
10. Нарисовать самолет, который может при посадке выпускает шасси. Родительский класс – перемещающийся прямоугольник.

11. Нарисовать домик, в котором открываются двери и окна. Родительский класс – перемещающийся прямоугольник.

12. Нарисовать паровоз, который выпускает дым. Родительский класс – перемещающийся прямоугольник.

13. Нарисовать воздушный шарик, который может лопнуть. Родительский класс – перемещающийся эллипс

14. Нарисовать лифт, который доставляет людей на нужный этаж. Родительский класс – перемещающийся прямоугольник.

15. Нарисовать тележку, на которой перевозят различные грузы. Родительский класс – перемещающийся прямоугольник.

Задание 2.

Реализовать иерархию классов, представляющих плоские геометрические фигуры. Необходимые фигуры: треугольник, круг, ромб, квадрат (отдельный класс), прямоугольник, звезда (пятиконечная, шестиконечная, восьмиконечная), шестиугольник и **любая другая фигура не основанная на вышеперечисленных**. Каждая фигура должна позволять производить следующие операции (при этом операции в общем виде для произвольной геометрической фигуры можно не реализовывать, реализовать их нужно для каждого конкретного типа фигур):

- нахождение площади и периметра, центра масс (для многоугольников можно воспользоваться триангуляцией);
- изменение координат центра масс (при этом вся фигура сдвигается на столько же, на сколько сдвинут ее центр);
- управление параметрами, специфическими для конкретных типов фигур (координаты вершин многоугольника, радиусы эллипса и т. п.);
- осуществление базовых преобразований: перемещение, поворот относительно точки, масштабирование относительно точки; при этом фигура должна изменяться плавно в течение заданного промежутка времени и на каждый такт изменения генерировать событие (можно воспользоваться Timer);
- возможность рисования фигуры на заданной канве (Canvas).

Каждый класс необходимо помещать в отдельный модуль (unit).

Создать оконное приложение, которое позволит пользователю создавать фигуры и производить с ними вышеуказанные операции.

Обязательной является реализация многоуровневой иерархии, то есть когда одни фигуры являются разновидностями других (стандартный плохой пример: прямоугольник – разновидность многоугольника, а квадрат –

разновидность прямоугольника. Пример плохой т.к. принцип инвариантности не соблюден).

Задание 3. (здесь можно получить 4)

Разработайте класс Date, хранящий день, месяц, год.

Определите и реализуйте функции-члены этого класса ():

- *Date NextDay()* – определить дату следующего дня;
- *Date PreviousDay()* – определить дату предыдущего дня, *bool IsLeap()* – является ли год этой даты високосным, *short WeekNumber()* – номер недели в году для текущей даты;
- *int DaysTillYourBithday(Date bithdaydate)* – сколько дней до вашего дня рождения от текущей даты;
- *int Duration(Date date)* – количество дней между текущей и данной датой.

Далее необходимо выполнить следующее задание: дан текстовый файл с датами (01.01.0001 24.04.2000 29.02.2000 31.12.2021). Прочитать данные в динамический массив объектов класса и для каждого элемента коллекции вычислить а) следующий день, б) разницу между текущим и следующим элементом коллекции. С помощью QT вывести на экран информацию о датах в файле таблицей. По кнопке вызывать функции класса и выводить информацию на экран. Текущий день вычислять через время из системы или используя стандартные функции. Добавить в форму поле для ввода даты рождения пользователя. Добавить функцию изменения и добавления дат в файл без полной перезаписи и соответствующий функционал в формах. Добавить функцию открытия .txt файла из любого дискового пространства.

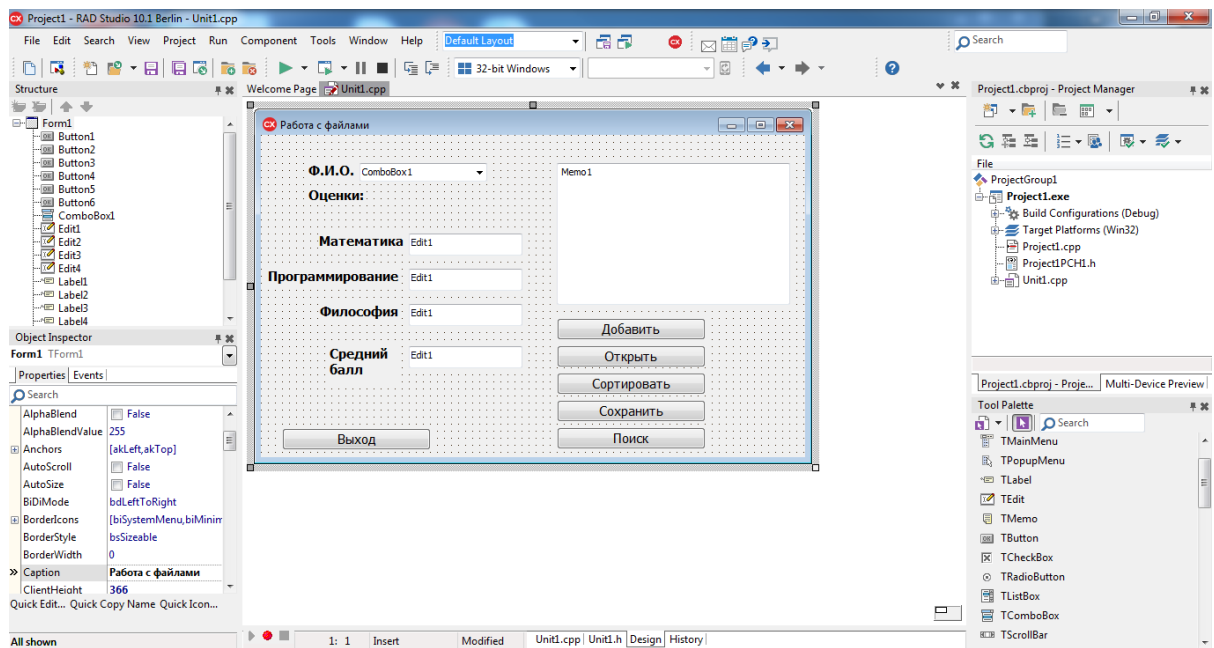
Задание 4. (здесь можно получить 6)

Разработать программу в соответствии с вариантом, реализующую работу с файлами в приложении с визуальными компонентами: Edit, Button, ComboBox, SaveDialog, OpenFileDialog, Memo, компонентного класса FileStream, методов LoadFromFile и SaveToFile.

Информация, обрабатываемая программой, должна храниться в файле. Использовать динамический массив объектов класса, который содержит данные в соответствии с заданием. Класс определить в отдельном модуле проекта (Add/Unit).

В каждом варианте задания реализовать следующие функции класса: инициализации данных класса, добавления, удаления, корректировки(сортировки) и просмотра записей файла.

Примерный вариант интерфейса:



Индивидуальные задачи по вариантам:

1. Магазин имеет список товаров на своем складе. Каждый пункт списка содержит следующую информацию о товаре: группа товаров, код товара, наименование продукции, наименование модели, цена, количество. Требуется:

- отсортировать товары внутри группы по одному из признаков: код товара, наименование продукции, цена;
- вывести товары, количество которых на складе меньше заданного значения;
- осуществлять поиск товара по наименованию продукции и модели.

2. Написать программу формирования ведомости об успеваемости студентов. Каждый пункт этой ведомости должен содержать: наименование специальности, номер группы, ФИО студента, оценки, полученные им за последнюю сессию, средний балл (вычисляется в программе). Требуется:

- отсортировать весь список студентов в порядке убывания среднего балла;
- отсортировать студентов в группе в порядке убывания среднего балла;
- вывести список студентов, не получивших в сессию неудовлетворительных оценок;
- осуществлять поиск студента по ФИО.

3. В радиоателье хранятся квитанции о сданной в ремонт радиоаппаратуре. Каждый пункт содержит следующую информацию: наименование группы изделий (телевизор, радиотелефон и т. п.), марка изделия, дата приемки в ремонт, дата исполнения заказа, состояние готовности заказа (выполнен или не выполнен). Требуется:

- выдать информацию о состоянии готовности заказов на текущие сутки по группам изделий;
- выдать информацию о заказах, невыполненных в срок;
- отсортировать заказы в группах по убыванию даты исполнения заказа;
- осуществлять поиск изделия по наименованию, дате приемки, дате исполнения заказа.

4. Разработать программу («электронную сваху») для службы знакомств. Имеется 2 списка: список женихов и список невест. В каждом списке кандидат (жених или невеста) характеризуется структурой, содержащей следующие поля: порядковый номер кандидата, данные о кандидате (имя, возраст, рост, вес, привычки, хобби), требования к партнеру (в виде диапазона Min-Max для возраста, роста, веса). Требуется:

- выявить все возможные пары с учетом требований кандидатов;
- при согласии сторон пара считается сформированной и кандидаты в списке помечаются как удаленные.

5. В фирме имеется информация о комплектующих изделиях компьютерной техники с указанием типа комплектующего изделия, фирмы-изготовителя, модели, параметров, цены, информации о наличии. Требуется:

- подобрать все возможные варианты комплектации компьютера в заданном ценовом диапазоне;
- осуществлять поиск изделия по типу изделия, фирме-изготовителю, модели;
- отсортировать изделия для каждого типа по убыванию цены.

6. В магазине имеется список поступивших в продажу CD/DVD дисков. Каждый пункт списка содержит: тип хранимой информации (фильм, музыка, СОФТ и т. п.), наименование, автора, цену и примечание. Требуется:

- сортировать внутри каждого типа информацию по наименованию либо по автору;
- осуществлять поиск диска по автору, по наименованию;
- осуществлять выбор информации по типу, по автору.

7. В деканате имеется список студентов. Каждый пункт этого списка содержит: номер группы, ФИО студента, оценки, полученные им за учебный год (две сессии: зимнюю и летнюю), средний балл по каждой сессии (вычисляется в программе). Требуется:

- сформировать список задолженников по результатам зимней сессии;
- сортировать списки по ФИО студентов внутри каждой группы;
- сформировать список студентов для отчисления (получивших две и более неудовлетворительные оценки за летнюю сессию и имевших задолженности (хотя бы одну неудовлетворительную оценку) за зимнюю сессию);
- осуществлять поиск студента в исходном списке по ФИО.

8. В больнице имеется общий список больных. Каждый пункт списка содержит: ФИО больного, пол больного, возраст, диагноз, номер палаты (при добавлении поле пустое). Требуется:

- разместить больных по палатам так, чтобы больные с одинаковым диагнозом располагались, по возможности, в одной палате. При этом **разнополых больных в одну палату размещать нельзя**. После размещения выдать список больных, которых не удалось разместить в палаты, с указанием ФИО больного, пола больного, возраста, диагноза;
- осуществлять поиск больных по номеру палаты, диагнозу, ФИО.

9. В проектной фирме есть список работ, закрепленный за сотрудниками. Каждый пункт списка содержит: название проекта, задание в рамках данного проекта, ФИО исполнителя, ФИО руководителя, дату выдачи задания, срок выполнения, дату сдачи задания сотрудником. Требуется:

- выдать список всех проектов руководителя;
- выдать список всех задач сотрудника;
- выдать список всех сотрудников проекта;
- выдать список всех сотрудников, не справившихся с заданием в срок.

10. В проектной фирме есть список работ, выполняемых сотрудником. Каждый пункт списка содержит: ФИО исполнителя, название проекта, задание в рамках данного проекта, дату выполнения задания, даты начала и окончания работы. Требуется:

- выдать список всех проектов фирмы, отсортированных по дате завершения работы;
- выдать список всех задач, работа над которыми велась одним сотрудником за текущие сутки;
- выдать список сотрудников с указанием суммарного времени работы каждого за прошедший месяц.

11. В проектной фирме есть список работ, закрепленный за сотрудниками. Каждый пункт списка содержит: название проекта, задание в рамках данного проекта, ФИО исполнителя, ФИО руководителя, дату выдачи задания, срок выполнения. Требуется:

- выдать список всех задач по конкретному проекту и их исполнителей;
- выдать список всех задач, срок завершения выполнения которых – ближайший месяц.

12. Дана рейтинговая таблица футболистов, содержащая ФИО футболиста, клуб, амплуа, количество забитых мячей, сумму штрафных очков. Требуется:

- выдать список из 10 самых успешных игроков текущего сезона, имеющих максимальное количество забитых мячей и минимальное количество штрафных очков;
- выдать список из 10 игроков с максимальной суммой штрафных очков;

- отсортировать список по убыванию количества забитых мячей.

13. Разработать программу «Биржа труда». Имеется список фирм с вакансиями. Каждый пункт списка содержит: название фирмы, должность, оклад, количество дней отпуска, требования к нанимаемому (наличие высшего образования (да/нет), возрастной диапазон (min/max); работа в данной должности не менее N-лет). Также имеется список кандидатов, каждый пункт которого содержит: ФИО кандидата, дату рождения, наличие высшего образования (да/нет), желаемую должность, минимальный оклад, список должностей, занимаемых ранее, с периодом работы. Требуется:

- для каждой фирмы подобрать возможных кандидатов по каждой вакансии;
- выдать список дефицитных должностей (кандидаты отсутствуют);
- выдать список кандидатов, для которых не найдена вакансия.

14. Разработать программу «Биржа труда». Имеется список фирм с вакансиями. Каждый пункт списка содержит: название фирмы, наименование специальности, должность, оклад, количество дней отпуска, требования к нанимаемому (наличие высшего образования (да/нет), возрастной диапазон (min/max)). Также имеется список кандидатов, каждый пункт которого содержит: ФИО кандидата, дату рождения, специальность, наличие высшего образования (да/нет), желаемую должность, минимальный оклад. Требуется:

- для каждого кандидата подобрать список возможных вакансий;
- выдать список дефицитных вакансий (кандидаты отсутствуют);
- выдать список фирм, для которых не подобран ни один кандидат.

15. В ресторане имеется меню с указанием названия блюда, его описания и цены. Также имеется список заказов за текущие сутки. Каждый пункт списка содержит: номер заказа, номер столика, название блюда, количество порций. Требуется:

- оформить счет по каждому заказу;
- выдать список блюд ресторана, пользующихся максимальным спросом;
- отсортировать список блюд по возрастанию цены.

16. В ресторане имеется меню с указанием названия блюда, его категории (холодные закуски, супы и т. п.) и цены. Также имеется список заказов за текущие сутки. Каждый пункт списка содержит: номер заказа, номер столика, название блюда, количество порций. Требуется:

- определить самое «популярное» блюдо в категории;
- определить самый «прибыльный» заказ;
- выдать список заказов по убыванию суммы заказа.

17. На фирме имеется список заказов на покупку товаров на следующие сутки. Каждый пункт списка содержит: номер заказа, адрес доставки, дата и время (от – до) доставки, вес груза в килограммах. Также имеется список курьеров, каждый пункт которого содержит: номер курьера, ФИО курьера, время работы (от – до), грузоподъемность автомобиля. Требуется:

- распределить заказы между курьерами;
- выдать список всех заказов курьера в последовательности их выполнения;
- выдать список всех заказов, которые не могут быть исполнены в срок.

18. На фирме формируются списки заказов на покупку товаров. Каждый пункт списка заказов содержит: номер заказа, дату заказа, реквизиты заказчика. Каждый заказ ссылается на список товаров в заказе, который в свою очередь содержит: номер заказа, код товара, количество товара. В прайс-листе хранятся код товара, цена за единицу товара, наименование товара. Требуется сформировать накладную по каждому заказу. Накладная содержит список товаров конкретного заказа с указанием количества и цены, а также суммарную стоимость заказа. Номер заказа и суммарная стоимость указывается в «шапке» накладной.

19. В поликлинике генерируется список талонов к врачу. Каждый пункт списка содержит: дату, время, номер очереди, ФИО больного (изначально поле пустое), номер кабинета, ФИО врача. Генерация талонов происходит на неделю, начиная с введенной даты, в соответствии с графиком работ врачей. График работы врача содержит: специализацию врача, ФИО врача, временной диапазон работы на каждый день с понедельника по субботу (длительность приема врачом больного – 15 минут). Требуется:

- сформировать списки талонов к врачу;
- осуществлять поиск всех записей к врачу на конкретную дату;
- осуществлять поиск записей о больном по ФИО.

20. В библиотеке имеется список книг. Каждый пункт списка содержит: фамилию автора (или авторов), название книги, год издания, издательство, количество страниц. Требуется:

- определить, имеются ли в данном списке книги, в названии которых встречается некоторое ключевое слово (например «программирование»);
- осуществлять поиск книги по ФИО автора либо по названию;
- отсортировать все книги библиотеки по году издания;
- выдать все книги одного издательства.

21. В магазине имеется список поступивших в продажу автомобилей. Каждый пункт списка содержит: страну-изготовитель, марку автомобиля, параметры автомобиля (тип двигателя, стоимость, расход бензина на 100 км, надежность (число лет безотказной работы), комфортность (в баллах). Покупатель, в свою очередь, имеет ряд требований по каждому из этих параметров. Эти требования могут задаваться в виде некоторого интервала (например, стоимость – 10...30 тыс. (\$)) либо выбираться из перечисленных списков. Необходимо:

- осуществлять поиск автомобилей, удовлетворяющих требованиям покупателя;

- отсортировать автомобили по маркам, а внутри списка одной марки по любому выбранному покупателем параметру.

22. В предвыборной кампании производится регистрация кандидатов в депутаты. Каждый кандидат при регистрации, указывает номер округа, в котором он собирается баллотироваться, ФИО, наименование партии, которую он представляет, возраст, профессию, доход за прошедший год. Требуется:

- сформировать информационный бюллетень, в котором приводится следующая информация по кандидатам от каждой политической партии: число поданных заявлений на регистрацию, средний возраст кандидатов, наиболее часто встречающаяся профессия и средний доход кандидатов;
- осуществлять поиск полной информации по кандидатам каждой партии.

23. В технической службе аэропорта имеется справочник, имеющий следующую структуру: тип самолета, год выпуска, расход горючего на 1000 км. Для определения потребности в горючем техническая служба запрашивает расписание полетов на следующие сутки. Каждый пункт расписания содержит: номер рейса, пункт назначения, дальность полета, тип самолета, время вылета, время приземления в пункте назначения. Требуется:

- рассчитать суммарное количество горючего, необходимое для обеспечения полетов на следующие сутки;
- осуществлять поиск всех рейсов в конкретный пункт назначения;
- составить документ-расписание полетов.

24. В библиотеке имеется список книг. Каждый пункт списка содержит: код книги, фамилию автора, название книги, год издания, язык издания. Также имеется список читателей, каждая запись которого содержит: код читателя, ФИО читателя, домашний адрес, контактный телефон. Книги, взятые каждым читателем, заносятся в отдельный список, содержащий код читателя, код книги, дату выдачи, требуемый срок возврата и реальную дату возврата книги. Требуется:

- создать список книг, находящихся у читателей;
- создать список должников (книгу не вернули в течение десяти дней после срока возврата);
- осуществлять поиск книг по ФИО автора либо по названию;
- осуществлять поиск читателя по ФИО.

25. У администратора железнодорожных касс хранится информация о свободных местах в поездах по всем направлениям на ближайшую неделю. Данная информация представлена в следующем виде: дата отправления, номер рейса, конечный пункт назначения, время отправления, число купейных и плацкартных мест, число свободных купейных и плацкартных мест. Требуется:

- выдать информацию об имеющихся свободных местах по каждому рейсу и каждому типу мест;
- выдать документ-расписание движения поездов по каждому дню недели;

- выдать список всех рейсов в пункт назначения с указанием общего числа свободных мест и числа проданных билетов.

26. Имеется ведомость абитуриентов, сдавших вступительные экзамены в институт. В каждой строке этой ведомости записана ФИО абитуриента, специальность, на которую он поступает, средний балл аттестата, полученные оценки по отдельным дисциплинам (например, физика, математика, русский язык). Требуется:

- выдать информацию по каждой специальности, содержащую ФИО абитуриента, суммарный балл каждого из них;
- отсортировать записи по каждой специальности по убыванию суммарного балла;
- осуществлять поиск абитуриента по ФИО.

27. В больнице имеется общий список больных. Каждый пункт списка содержит: ФИО больного, пол больного, возраст, диагноз, номер палаты, дату поступления, дату выписки. Требуется:

- указать номера палат, в которых лежат больные с более чем тремя разными диагнозами;
- сортировать списки не выписавшихся больных по убыванию даты поступления;
- осуществлять поиск больных по номеру палаты, полу, диагнозу, ФИО, возрасту (в режиме диапазона min – max).

28. Разработать программу («электронную сваху») для службы знакомств. Имеется один список женихов и невест. В списке кандидат (жених или невеста) характеризуются записью, содержащей следующие поля: порядковый номер кандидата, данные о кандидате (пол, ФИО, возраст, рост, вес), требования к партнеру (в виде диапазона min – max для возраста, роста, веса). Требуется:

- выявить все возможные пары;
- осуществлять поиск всех возможных кандидатов;
- вывести отсортированный по убыванию возраста список для кандидатов-женихов и кандидатов-невест.

29. Разработать программу формирования ведомости об успеваемости студентов. Каждый пункт этой ведомости должна содержать: наименование специальности, номер группы, ФИО студента, форму обучения (бюджетная, платная), оценки, полученные им в сессию, средний балл, вычисляемый программно. Требуется:

- отсортировать в группах студентов по ФИО;
- вывести список студентов, которые учатся на «отлично» (имеют 9- и 10-балльные оценки) с формой обучения «платная»;
- вывести списки студентов в разрезе формы обучения в порядке убывания среднего балла;
- осуществлять поиск студентов по ФИО, номеру группы, форме обучения.

30. На складе комплектующих имеется список деталей, каждая из которых характеризуется принадлежностью к группе товаров, наименованием товара, количеством, рядом признаков: признак 1 (числовой), признак 2 (строковый), признак 3 (логический). Требуется:

- отсортировать товары в группе по убыванию признака 1;
- выдавать информацию по запросу о количестве имеющихся деталей по каждому наименованию, о количестве деталей указанного наименования с заданными признаками, о количестве деталей различных наименований, имеющих одинаковые один, два или три признака.

Задание 5. (здесь можно получить 8)

Необходимо разработать иерархию классов Expression для представления арифметических выражений. Конкретнее, вам нужно определить три класса: Expression — базовый класс иерархии, Number — для представления чисел и BinaryOperation — класс описывающий бинарную операцию (+, --, * или /).

Класс Number должен хранить значение типа double.

Класс BinaryOperation должен хранить указатель на левый и правый операнды, которые сами являются арифметическими выражениями, а также тип операции (+, --, * или /), которую нужно над ними произвести.

Во всех классах должен быть метод evaluate, который возвращает значение типа double — значение соответствующего арифметического выражения, например, значение экземпляра типа Number — это число, которое он хранит, а если у вас есть объект BinaryOperation с операцией +, то нужно вычислить значения левого и правого операнда и вернуть их сумму.

В данном задании вам нужно расставить ключевое слово virtual там, где это необходимо, определить метод evaluate там, где его не хватает, а также реализовать деструкторы, там где они нужны.

При выполнении этого задания учтите, что при уничтожении объекта BinaryOperation он отвечает за уничтожение левого и правого операндов (гарантируется, что они выделены в динамической памяти).

Например, выражению $3 + 4.5 * 5$ может соответствовать следующий код:

```
// сначала создаём объекты для подвыражения 4.5 * 5
Expression * sube = new BinaryOperation(new Number(4.5), '*', new
Number(5));

// потом используем его в выражении для +
Expression * expr = new BinaryOperation(new Number(3), '+', sube);

// вычисляем и выводим результат: 25.5
```

```
std::cout << expr->evaluate() << std::endl;

// тут освобождаются *все* выделенные объекты

// (например, sube будет правым операндом expr, поэтому его удалять не
нужно)

delete expr;
```

Требования к реализации: при выполнении этого задания не нужно вводить или выводить что-либо. Вы можете заводить любые вспомогательные функции, методы или классы, но не нужно реализовывать функцию main.

Пример **начальной** реализации в Приложении 1.

(Задание повышенной сложности на 9-10 баллов) Предполагаемое решение этого задания не переносимо с точки зрения стандарта, **однако** оно проверено на различных версиях компиляторов g++/clang++/msvc.

Вам требуется реализовать функцию, которая принимает на вход два указателя на базовый класс Expression, и возвращает true, если оба указателя указывают на самом деле на объекты одного и того же класса, и false в противном случае (т.е. если оба указателя указывают на BinaryOperation, то возвращается true, а если один из них указывает на Number, а второй на BinaryOperation, то false).

```
bool check_equals(Expression const *left, Expression const *right) {}
```

Требования к реализации: пользоваться typeid и dynamic_cast запрещено. Вызывать методы по переданным указателям запрещено.

Приложение 1

```

struct Expression
{
    double evaluate() const = 0;
};
struct Number : Expression
{
    Number(double value) : value(value) {}
private:
    double value;
};
struct BinaryOperation : Expression
{
    //Здесь op это один из 4 символов: '+', '-', '*' или '/', соответствующих операциям,
    //которые вам нужно реализовать.
    BinaryOperation(Expression const * left, char op, Expression const * right)
        : left(left), op(op), right(right) {}
private:
    Expression const * left;
    Expression const * right;
    char op;
};

```