Правительство Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Кафедра «Компьютерная безопасность»

ОТЧЕТ
К ЛАБОРАТОРНОЙ РАБОТЕ №4

по дисциплине

«Языки программирования»

Работу выполнил

студент группы СКБ213 _____ Д.А. Мишкина

подпись, дата

Работу проверил С.А. Булгаков

_____

подпись, дата

Москва 2022

Содержание

Постановка задачи

Ответвление

Необходимо создать ответвление (fork) моего репозитория Lab-02 (название оставить без изменения, а видимость поставить приватной). Назначить меня в новом репозитории соавтором с правами администратора. Вики, Задачи а также Проекты отключить.

В локальной копии создавать ветки (branch) по необходимости.

Этап разработки

Доработать программу "Текстовый редактор" из задания лабораторной работы 3 следующим образом:

* Добавить в окно "Граф вызовов" главное меню "Файл" с пунктами "Сохранить" и "Закрыть". При выборе "Сохранить" открывается диалоговое окно с выбором файла сохранения графа в формате XML.

Разработать вспомогательную программу содержащую главное меню "Файл" с пунктами "Открыть" и "Выход". При выборе "Открыть" открывается диалоговое окно с выбором файла в формате XML для чтения графа и отображения его в окне программы.

По разработанной программе подготовить отчет в соответствии с требованиями ГОСТ 19. Отчет в обязательном порядке должен содержать диаграмму классов в формате UML-2.0. Отчет включить в состав исходных кодов программы в виде файла формата PDF.

Запрос слияния

По итогу выполнения работы создать запрос на слияние (pull request) и назначить меня рецензентом.

Общий алгоритм решения поставленной задачи

В первую очередь необходимо добавить главное меню в окно с графом вызовов. После этого при открытии и сохранении графа вызовов все необходимые данные со сцены записываются в XML файл.

Класс XmlWindow реализует окно с меню, которое открывается при запуске программы. При выборе пункта меню “File” -> “Open” открывается файл в формате XML, и с помощью данных из этого файла элементы добавляются на сцену.

1 Сохранение текстовых данных

При составлении графа в CallGraph::drawGraph(QFile input_file) к текстовому элементу сцены добавляются текстовые данные data, которые позволяют в дальнейшем успешно вывести текст вершин сохранённого графа.

2 Сохранение в XML формат

Помимо самого графа вызовов класс CallGraph реализует возможность сохранения графа и закрытия программы. При сохранении данные записывается в формат XML следующим образом: пользователь нажимает на соответствующий пункт меню и выбирает файл, в который хочет сохранить (с помощью фильтров реализовано, чтобы открыть можно было только файл нужного формата), после этого создаётся объект типа QXmlStreamWriter, программа проходит по всей сцене с помощью цикла foreach. В случае, если текущий элемент является текстом, то в XML файл записываются координаты и сам текст. Если текущий элемент - это прямоугольник, то

записываются его координаты, ширина и длина, что реализуется с помощью метода boundingRect(). Если же текущий элемент - линия, то проверяется, если координата Y конца больше, чем координата Y начала, то координата Y начала становится равной координате Y описывающего данную линию прямоугольника, иначе - сумме координаты Y и высоты. Далее в XML файл записываются координаты начала и конца линии. После сохранения файла название окна становится равным названию файла.

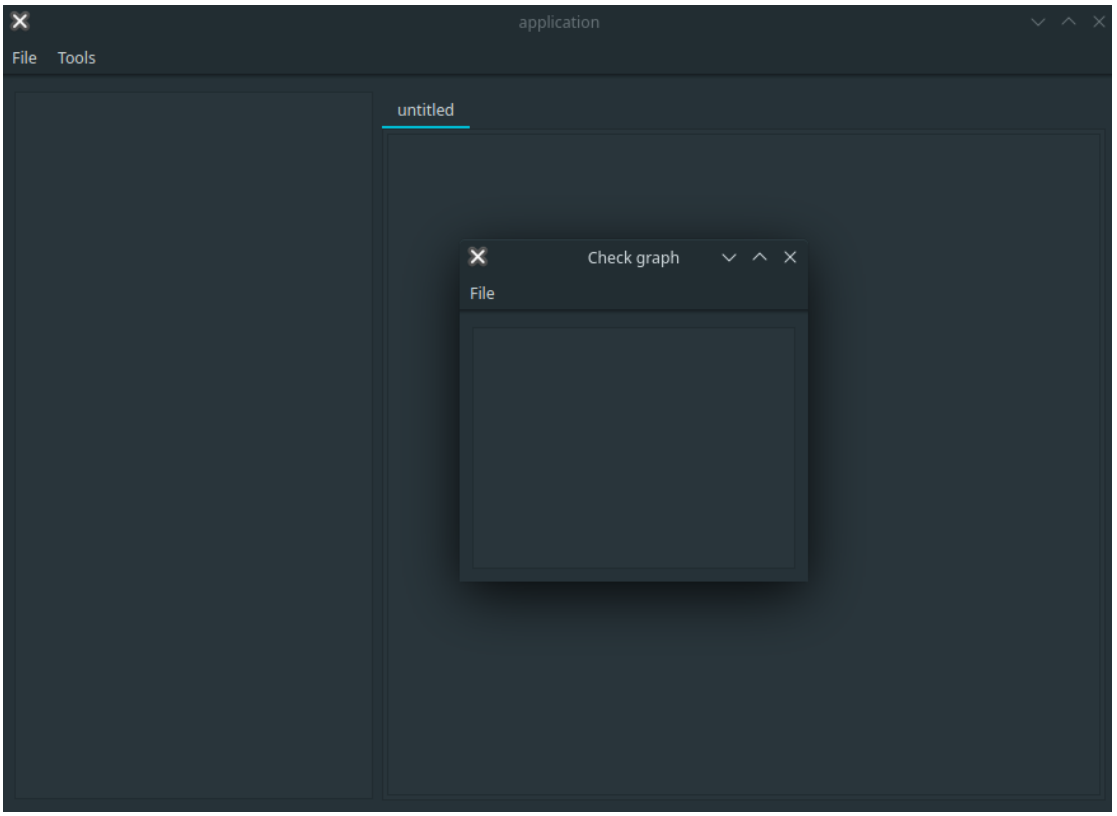
3 Чтение из XML формата

Класс XmlWindow реализует окно с меню и сценой. С помощью меню можно открыть файл в формате XML, что реализуется с помощью фильтров, и закрыть данное окно. При открытии программа проходит циклом по всему файлу. Внутри цикла в каждой строке тега GraphicsItem программа записывает данные атрибутов в переменные. Если тип соответствует линии, то на сцену добавляется линия с координатами из файла. Если прямоугольнику, на сцену добавляется прямоугольник с координатами, шириной и длиной из файла. Если же тип - текст, то текст из data устанавливается в нужные координаты. После открытия файла название окна становится равным названию открытого файла.

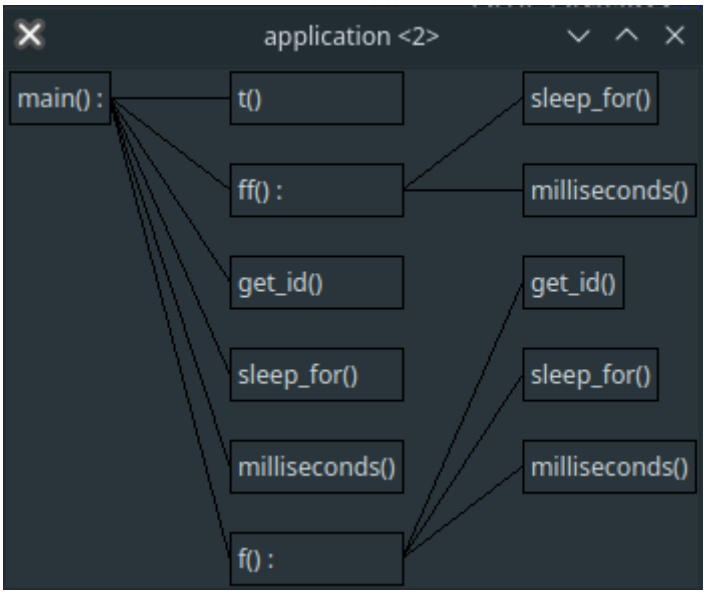
Тестирование

1 Начальное состояние программы

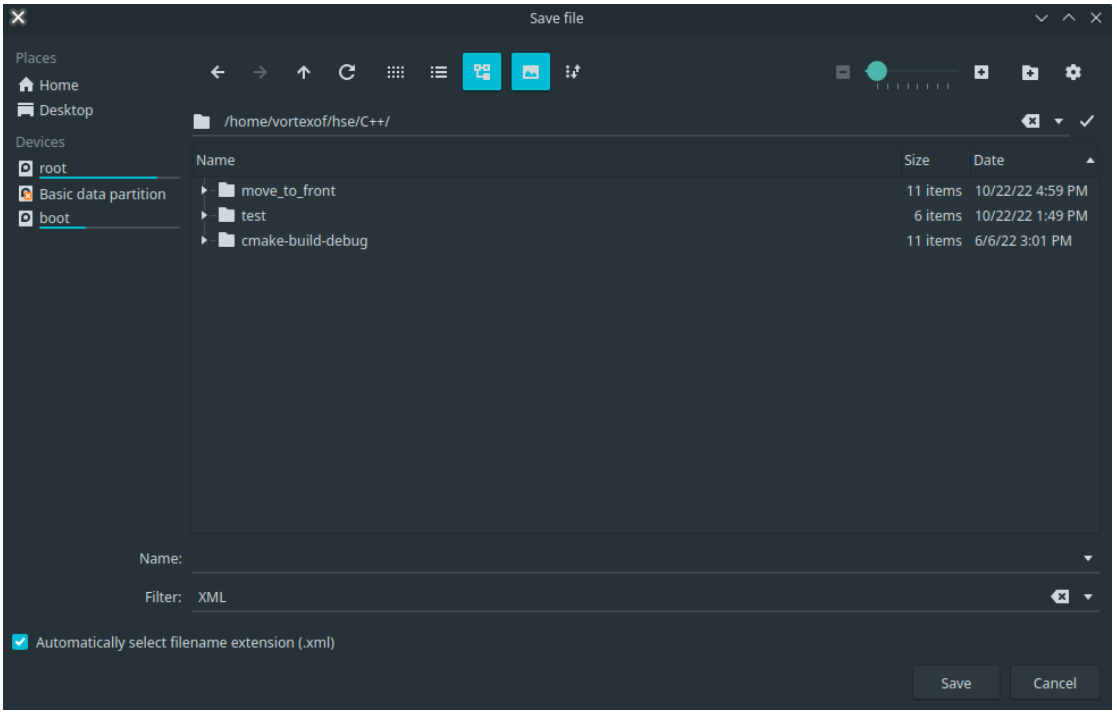
При запуске программы открывается два окна.



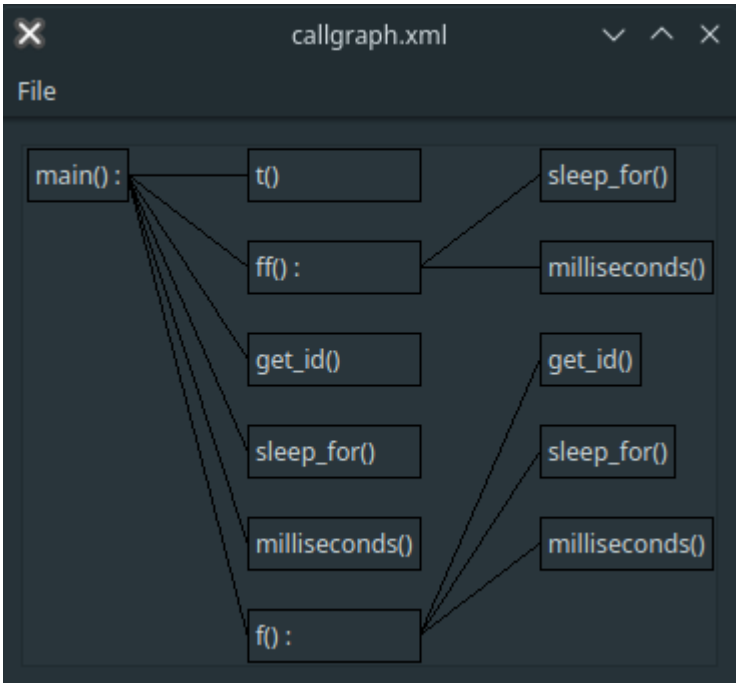
2 Отрисованный граф вызовов



3 Сохранение файла графа вызовов



4 Сохранённый граф вызовов

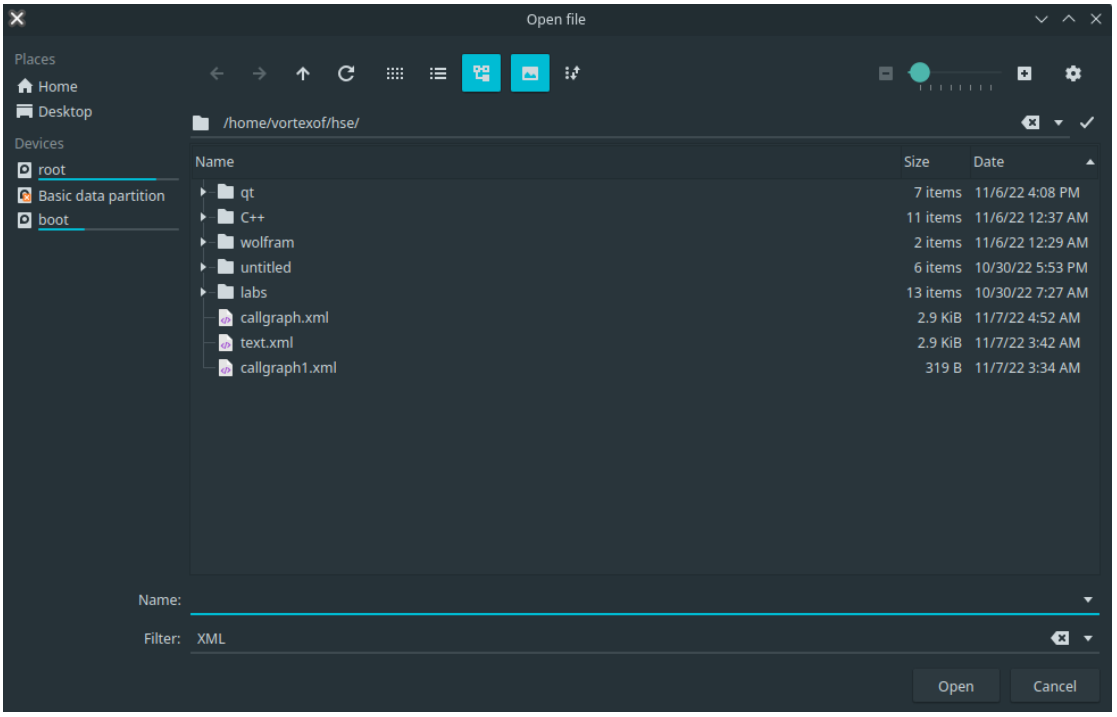


5 XML файл

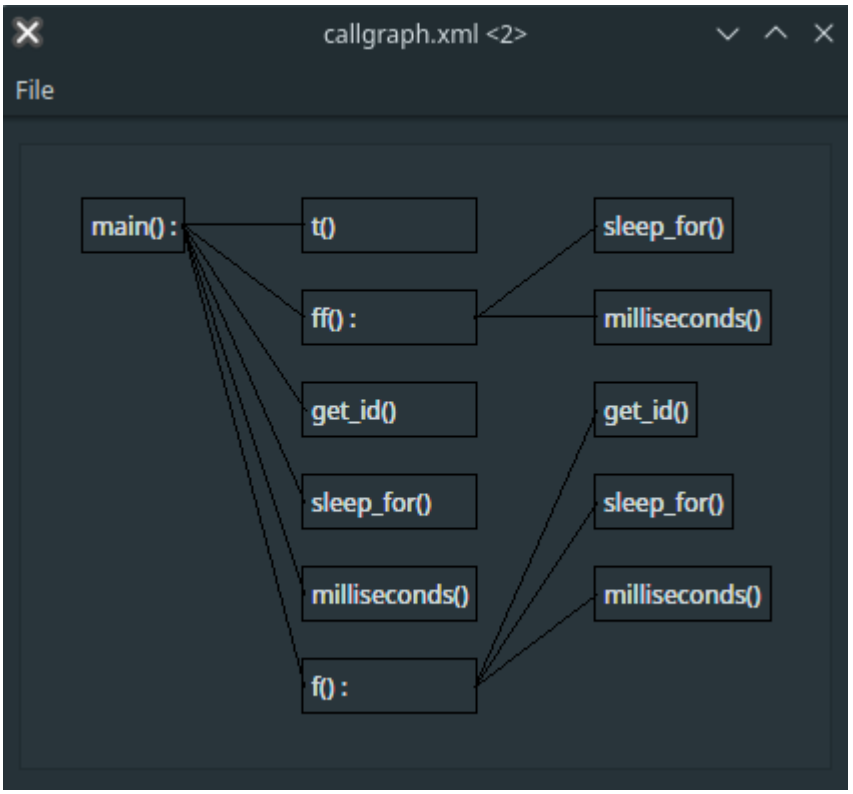
XML файл данного графа вызовов.

```
▼<SceneData version="v1.0">
  ▼<GraphicsItemList>
     <GraphicsItem type="6" x1="255.299" y1="289.701" x2="316.701" y2="243"/>
     <GraphicsItem type="3" x="315.5" y="229.5" width="87.625" height="27"/>
     <GraphicsItem type="8" x="316" y="230" text=" milliseconds()"/>
     <GraphicsItem type="6" x1="255.308" y1="289.692" x2="316.692" y2="197"/>
     <GraphicsItem type="3" x="315.5" y="183.5" width="68.8906" height="27"/>
     <GraphicsItem type="8" x="316" y="184" text=" sleep_for()"/>
     <GraphicsItem type="6" x1="255.342" y1="289.658" x2="316.658" y2="151"/>
     <GraphicsItem type="3" x="315.5" y="137.5" width="51.4063" height="27"/>
     <GraphicsItem type="8" x="316" y="138" text=" get_id()"/>
     <GraphicsItem type="6" x1="255.5" y1="104.5" x2="316.5" y2="105"/>
     <GraphicsItem type="3" x="315.5" y="91.5" width="87.625" height="27"/>
     <GraphicsItem type="8" x="316" y="92" text=" milliseconds()"/>
     <GraphicsItem type="6" x1="255.299" y1="105.701" x2="316.701" y2="59"/>
     <GraphicsItem type="3" x="315.5" y="45.5" width="68.8906" height="27"/>
     <GraphicsItem type="8" x="316" y="46" text=" sleep_for()"/>
     <GraphicsItem type="6" x1="109.39" y1="58.39" x2="170.61" y2="289"/>
     <GraphicsItem type="3" x="169.5" y="275.5" width="87" height="27"/>
     <GraphicsItem type="8" x="170" y="276" text=" f() <void f () at /home/vortexof/hse/C++/main.cpp:6>:"/>
     <GraphicsItem type="6" x1="109.37" y1="58.3696" x2="170.63" y2="243"/>
     <GraphicsItem type="3" x="169.5" y="229.5" width="87" height="27"/>
     <GraphicsItem type="8" x="170" y="230" text=" milliseconds()"/>
     <GraphicsItem type="6" x1="109.342" y1="58.3421" x2="170.658" y2="197"/>
     <GraphicsItem type="3" x="169.5" y="183.5" width="87" height="27"/>
     <GraphicsItem type="8" x="170" y="184" text=" sleep_for()"/>
     <GraphicsItem type="6" x1="109.308" y1="58.3081" x2="170.692" y2="151"/>
     <GraphicsItem type="3" x="169.5" y="137.5" width="87" height="27"/>
     <GraphicsItem type="8" x="170" y="138" text=" get_id()"/>
     <GraphicsItem type="6" x1="109.299" y1="58.299" x2="170.701" y2="105"/>
     <GraphicsItem type="3" x="169.5" y="91.5" width="87" height="27"/>
     <GraphicsItem type="8" x="170" y="92" text=" ff() <void ff (string s) at /home/vortexof/hse/C++/main.cpp:11>:"/>
     <GraphicsItem type="6" x1="109.5" y1="58.5" x2="170.5" y2="59"/>
     <GraphicsItem type="3" x="169.5" y="45.5" width="87" height="27"/>
     <GraphicsItem type="8" x="170" y="46" text=" t()"/>
     <GraphicsItem type="3" x="59.5" y="45.5" width="51" height="27"/>
     <GraphicsItem type="8" x="60" y="46" text="main() <int main () at /home/vortexof/hse/C++/main.cpp:16>:"/>
  </GraphicsItemList>
</SceneData>
```

6 Открытие файла графа вызовов



7 Открытый граф вызовов



UML-диаграмма

Незаполненные участки диаграммы идентичны лабораторной работе №3.