

Задача 4. Декомпозиция программы – 2

В задаче «Декомпозиция программы» мы разбили монолитный код на набор функций и классов. Теперь мы сделаем ещё один шаг и разделим нашу программу на несколько файлов. В этой задаче вам нужно создать проект, состоящий из следующих файлов:

1. query.h, в него кладём:

- enum class QueryType
- struct Query
- объявление istream& operator >> (istream& is, Query& q)

2. query.cpp, в него кладём

- определение istream& operator >> (istream& is, Query& q);

3. responses.h:

- struct BusesForStopResponse
- ostream& operator << (ostream& os, const BusesForStopResponse& r)
- struct StopsForBusResponse
- ostream& operator << (ostream& os, const StopsForBusResponse& r)
- struct AllBusesResponse
- ostream& operator << (ostream& os, const AllBusesResponse& r)

4. responses.cpp: определения всего, что объявлено в responses.h

5. bus_manager.h: объявление класса BusManager

6. bus_manager.cpp: определения методов класса BusManager

7. main.cpp: функция main

Как отправлять проект на проверку

В заданиях по программированию Система позволяет отправлять на проверку только один файл, поэтому для того, чтобы сдать свой проект в тестирующую систему, вам надо запаковать все свои файлы в архив и отправить его на проверку. Поддерживаемые форматы архивов: rar, zip, 7z, tar, tar.gz, а также многие другие (для распаковки мы используем программу [dtrx](#), на странице с её описанием приведён полный список поддерживаемых форматов).

Как будет тестироваться ваше решение

Ниже вам дана заготовка для файла main.cpp, содержащая функцию main. Это точно такая же заготовка, которая была дана в задаче «Декомпозиция программы». Тестирование вашего решения будет выполняться в два этапа. На первом этапе автоматическая тестирующая система распакует присланный вами архив и соберёт извлечённые файлы в исполняемый файл. При этом функция main в вашем проекте будет заменена на ту, которая дана в заготовке файла main.cpp. Затем этот исполняемый файл будет запущен на наборе тестов. Тестирование выполняется так же, как и для большинства задач на нашем курсе: тест подаётся в stdin, замеряется время выполнения программы, а затем анализируется stdout.

На втором этапе будет выполняться тестирование отдельных файлов вашего проекта. Проверяется, что файл query.h действительно содержит перечислимый тип QueryType, что BusManager::GetBusesForStop возвращает корректно сформированный объект BusesForStopResponse т.д.

Заготовка файла main.cpp:

main.cpp