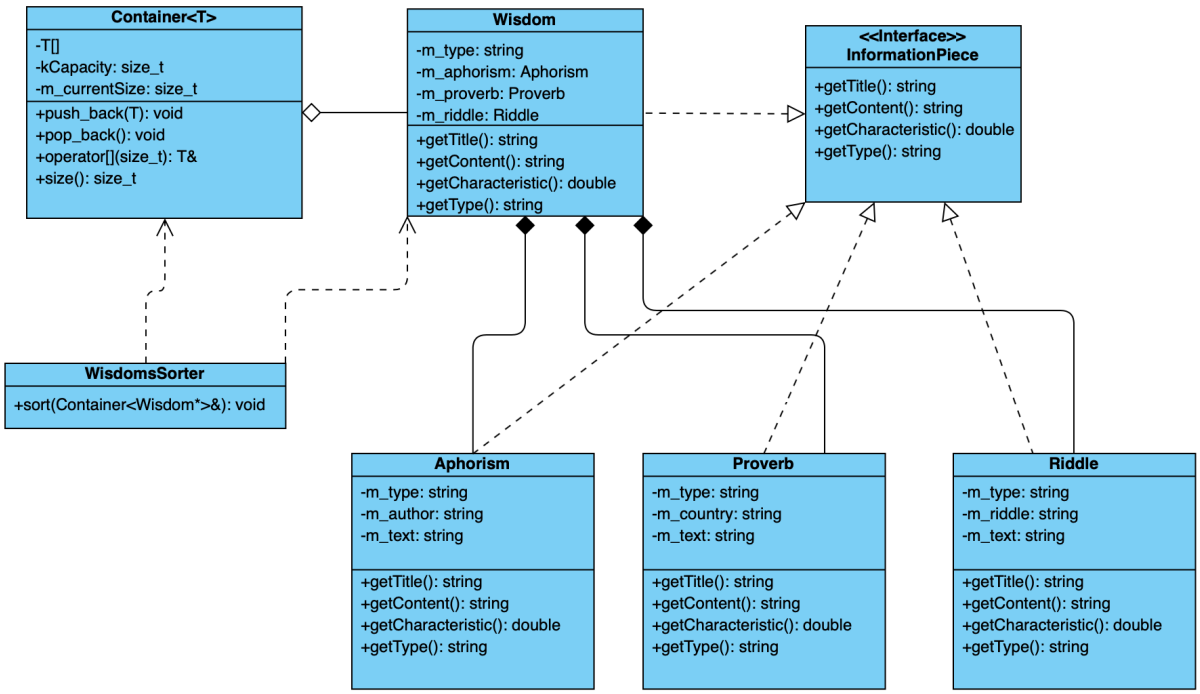
Задание 2: Вычислительная Система, ориентированная на объектно-ориентированный подход

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив)	Общие для всех альтернатив переменные	Общие для всех альтернатив функции
10. Кладезь мудрости.	1. Афоризмы (один из авторов – строка символов) 2. Пословицы и поговорки (страна – строка символов) 3. Загадки (ответ – строка символов)	Содержание кладези мудрости – строка символов	Частное от деления количества знаков препинания в содержательной строке на длину строки (действительное число)

Упорядочить элементы контейнера по возрастанию используя сортировку с помощью прямого включения (Straight Insertion). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

1) Структурная схема



Заголовочные файлы:
informationPiece.h
wisdom.h
aphorism.h
proverb.h
riddle.h
container.h
wisdomGenerator.h
wisdomsSorter.h
Файлы исходного кода:
informationPiece.cpp
wisdom.cpp

aphorism.cpp
proverb.cpp
riddle.cpp
wisdomGenerator.cpp
wisdomsSorter.cpp

2) Метрики
   В качестве основной метрики, определяющей характеристики программы будем использовать среднее арифметическое время работы программы на 10 000 случайных «Мудростях» (будем выполнять 5 тестов). Также будем сравнивать суммарный размер текстов.
3) Замеры характеристик
   1. Процедурный подход

```
(base) MacBook-Pro-1:cmake-build-debug a1$ ./task1 -r 10000 output5_before output5_after
Time difference = 128167669[microseconds]
(base) MacBook-Pro-1:cmake-build-debug a1$ ./task1 -r 10000 output5_before output5_after
Time difference = 133989889[microseconds]
(base) MacBook-Pro-1:cmake-build-debug a1$ ./task1 -r 10000 output5_before output5_after
Time difference = 132108060[microseconds]
(base) MacBook-Pro-1:cmake-build-debug a1$ ./task1 -r 10000 output5_before output5_after
Time difference = 121290507[microseconds]
(base) MacBook-Pro-1:cmake-build-debug a1$ ./task1 -r 10000 output5_before output5_after
Time difference = 131595796[microseconds]
```

   Среднее арифметическое: 129 430 384.2 microseconds
   Суммарный размер текстов: 12839 Б
   2. Объектно-ориентированный подход

```
(base) MacBook-Pro-1:cmake-build-debug a1$ ./avs2 -r 10000 output3_before output3_after
Time difference = 25791256[microseconds]
(base) MacBook-Pro-1:cmake-build-debug a1$ ./avs2 -r 10000 output3_before output3_after
Time difference = 24833082[microseconds]
(base) MacBook-Pro-1:cmake-build-debug a1$ ./avs2 -r 10000 output3_before output3_after
Time difference = 25214798[microseconds]
(base) MacBook-Pro-1:cmake-build-debug a1$ ./avs2 -r 10000 output3_before output3_after
Time difference = 25865687[microseconds]
(base) MacBook-Pro-1:cmake-build-debug a1$ ./avs2 -r 10000 output3_before output3_after
Time difference = 25292525[microseconds]
```

   Среднее арифметическое: 25 399 469.6 microseconds
   Суммарный размер текстов: 7954 Б

   Вывод: объектно-ориентированное решение занимает меньше места в виде исходных текстов и работает быстрее