
ОБЗОР ПРОГРАММЫ

Программа ΔTest предоставляет простой и универсальный способ тестирования. Программа поддерживает современный формат обмена данными JSON (JavaScript Object Notation), который удобен для чтения и написания не только машиной, но и человеком. Данное решение является идеальным для поставленной задачи, так как человек сможет писать свои тесты для ΔTest просто в блокноте или в специальном редакторе для JSON, главное, чтобы объект соответствовал поддерживаемому формату, который будет приведен ниже.

Программа рассчитана на пользователя с хорошим уровнем компьютерной грамотности, так как нужно умение работы в командной строке. Язык программы английский, но язык теста уже зависит уже от самого теста. То есть пользователю не обязательно владеть английским языком для использования программы, потому что он может загрузить в программу тест на любом языке.

СТРУКТУРЫ ДАННЫХ

В программе использованы следующие структуры данных:

1. Для хранения вопроса и вариантов правильных ответов

```
struct Question {
    Question(const std::string &question, const std::vector<std::string> &answers) :
        question(question), answers(answers) {}
    //Строка с вопросом
    std::string question;
    //Вектор из строк с вариантами правильных ответов
    std::vector<std::string> answers;
};
```

2. Для хранения названия темы и вопросов

```
struct Topic {
    Topic(const std::string &title, const std::vector<Question> &questions) :
        title(title), questions(questions) {}
    //Строка с названием темы
    std::string title;
    //Вектор из структуры вопросов с вопросами по данной теме
    std::vector<Question> questions;
};
```

3. Для хранения самого теста и работы с ним

```
class Quiz {
public:
    Quiz(const json &quizJson);
    void start();
private:
    void startQuizWithTopic(const Topic &topic);
    //Поле типа json для хранения json структуры данного теста
    json quizJson;
    //Вектор из структуры темы с темами данного теста
    std::vector<Topic> topics;
    //Строка с названием теста
    std::string name;
};
```

4. Формат JSON для хранения тестов в файле

```
{
  "name": "C++ quiz",
  "topics": [
    {
      "title": "Basic Concepts",
      "questions": [
        {
          "question": "Fill in the blanks to include the iostream header:\n#include
<----->",
          "answer": [
            "iostream"
          ]
        }
      ]
    }
  ]
}
```

Он из себя представляет объект, который включает себя в ключ "name", значением которого является название данного теста. Так же в объект включен массив объектов "topics", который в своем значении содержит упорядоченную коллекцию объектов с вопросами. Объекты вопросов содержат в себе ключи "question" и "answer". В значении ключа "question" лежит строка с самим вопросом, а "answer" в себе содержит массив строк с вариантами правильных ответов.

Парсинг JSON объекта происходит в конструкторе классе Quiz. Там происходит создание и инициализация структур Question и Topic.

АРХИТЕКТУРА СИСТЕМЫ

Сначала программа встречает пользователя рекламной вставкой с логотипом выдуманной компании A&K и логотипом самого ΔTest:

```
  _ _ _ _ _  _ _ _ _ _
 /   |   |   | /   /
 /  /|  |&|   /
 /   _ _ _ |   | \   \
/_/_/  |_ |  |_ | \_ \Inc.
Audacious & Keen
```

```
Presents:
-----
___/  |__  ___/-----/
___ /| |__ /  -  - \  ___/  /
_  /_ | | / /  /  __/(_  )/ /_
-----/  /  \___//___/ \___/
The universal testing application based on JSON
```

Работа с программой организована в виде навигации по различным пунктам меню. Пункты меню зависят от самого теста. Меню на примере демо-теста по C++:

```
Hello! Welcome to C++ quiz
This quiz has these topics:
1) Basic Concepts
2) Functions
0) Exit
Choose the topic:
```

Сверху показывается строка приветствия с названием теста, далее в меню предлагается пользователю выбрать по какой тебе он хочет пройти тест, в показанном случае по Основным понятиям и функциям C++.

```
1/11 question:
Fill in the blanks to include the iostream header:
#include <____>
Answer:
```

После выбора темы, сразу же на экран показывается первый вопрос. После Answer: пользователю предлагается ввести ответ на вопрос.

Wrong!

2/11 question:

Fill in the blanks to use names from std namespace:

_____ namespace std;

Answer:

```
2/11 question:
Fill in the blanks to use names from std namespace:
_____ namespace std;
Answer:
```

В зависимости от правильности ответа, на экране показывается соответствующая надпись и сразу же следующий вопрос.

You got 10 out of 11

После прохождения всего теста, пользователю выводится общая статистика, на основании которой он может уже делать выводы о знании данной темы.

СИСТЕМНЫЕ ТРЕБОВАНИЯ

Любое устройство под управлением macOS / Windows / GNU/Linux с установленным CMake для сборки программы под конкретную операционную систему. Аппаратная часть достаточна, если работают названные операционные системы.