

Требования к реализации учебно-демонстрационных программ

В зависимости от сложности темы курсовой проект выполняется либо индивидуально, либо коллективом разработчиков из 2-3 человек.

Общие требования

1. Учебно-демонстрационная программа (УДП) состоит из 3 подсистем:
 1. Подсистема просмотра учебного материала
 2. Подсистема демонстрации алгоритмов и операций
 3. Подсистема проверки знаний (тестирование)
2. Система должна обеспечивать расширение (дополнение) каждой из подсистем без существенной модификации уже реализованного функционала.
3. Функциональный код должен быть реализован на языке C++14 в объектно – ориентированном стиле с использованием стандартной библиотеки STL.
4. Интерфейс должен быть реализован на C++ с помощью одной из библиотек:
 1. Qt
 2. FLTK
 3. wxWidget
 4. GTK
 5. SFML
 6. Ultimate++

По согласованию с заказчиком могут быть использованы любые другие свободно распространяемые библиотеки C++ (например, Wt).

По согласованию с заказчиком могут быть использованы библиотеки, реализованные на JS (например, библиотека Sciter или аналогичные).

5. Интерфейсная часть системы не должна содержать функциональный код.
6. Система должна предоставлять интерактивную помощь по работе с ней.

7. Система должна быть протестирована с помощью одной из следующих систем модульного тестирования:

1. Googletest
2. Doctest
3. Catch

По согласованию с заказчиком могут быть использованы любые другие системы типа xUnit для C++ (например, Boost.Test)

Тестовый код должен быть описан в пояснительной записке в разделе описания программы в подразделе «Тестовый код».

8. В пояснительной записке должны быть показаны и описаны следующие виды диаграмм UML:

1. Диаграмма вариантов использования
2. Диаграммы последовательностей
3. Диаграммы классов
4. Диаграмма компонентов

Подсистема просмотра учебного материала.

1. Подсистема должна обеспечивать показ теоретического материала по теме.
2. Теоретический материал может включать:
 - a. Текст
 - b. Коды программ
 - c. Таблицы
 - d. Рисунки
3. Теоретический материал должен быть организован в виде гипертекста с разбиением на подразделы и страницы.
4. Должно быть обеспечено общее содержание. В содержании указываются подразделы теоретического материала (НЕ СТРАНИЦЫ).
5. Для каждой страницы должен быть обеспечен переход:
 - a. К следующей странице
 - b. К предыдущей странице
 - c. В начало подраздела
 - d. К общему содержанию.
6. Все определения, прописанные в теоретическом материале, должны быть собраны в отдельный словарь и в тексте теоретического материала снабжены гиперссылками.
7. Подсистема должна обеспечивать подключение новой порции теоретического материала без изменения программного кода.

Подсистема демонстрации алгоритмов и операций

1. Подсистема должна демонстрировать на экране выполнение всех операций и алгоритмов, относящихся к теме.
2. Каждый алгоритм должен демонстрироваться пошагово.
3. Подсистема должна обеспечивать пошаговый переход вперед и назад. Должна быть обеспечена возможность начать с первого шага, и выполнить без пошаговой демонстрации.
4. При демонстрации алгоритмов и операций должен быть доступен теоретический материал по соответствующему подразделу.

Подсистема тестирования

1. Подсистема должна обеспечивать тестирование пользователя в соответствии с установленными настройками.
2. Подсистема должна обрабатывать два типа вопросов:
 - a. Выбор нескольких ответов из нескольких альтернатив; количество от 5 до 9;
 - b. Открытый вопрос;
3. В текст вопроса может входить код программы.
4. Каждый вопрос характеризуется следующими атрибутами:
 - a. Тема к которой он относится (один вопрос может относиться к нескольким темам)
 - b. Трудность вопроса – минимальная, базовая, повышенная
 - c.
5. Вопросы должны быть записаны в текстовых файлах.
6. Подсистема должна обеспечивать возможность подключения новых файлов с вопросами без изменения программного кода.
7. Формат файла вопросов должен быть простой и подготовка его должна выполняться в простом текстовом редакторе типа Блокнот. Пользователю должна быть предоставлена инструкция по формированию файла вопросов. Формат файла вопросов должен быть описан в пояснительной записке.
8. Ответы должны сохраняться в бинарных файлах. Формат записи бинарного файла должен быть разработан и описан в пояснительной записке. Для каждого поля записи должно быть указано:
имя, тип, размер, назначение

Это в дополнительную карму

9. По согласованию с заказчиком подсистема должна обеспечивать настройки:
 - a. Количество вопросов, выбираемых для создания теста.
 - b. Список тем, из которых будут выбираться вопросы для тестирования.
 - c. Максимальный уровень сложности вопросов, включаемых в тест.
 - d. Время проведения теста
 - e. Способ окончания тестирования:
 1. Все вопросы отвечены.
 - ii. Закончилось время тестирования.
 - iii. Сделано k ошибок в ответах
 - iv. Доля неверных ответов превзошла D: $0 < D < 1$.
 - f. Контрольный / обучающий режим
10. В обучающем режиме должен быть доступен теоретический материал из соответствующего подраздела (темы). В контрольном режиме теоретический материал должен быть недоступен.
11. По согласованию с заказчиком должна быть разработана система оценивания
 - a. отдельного вопроса, которая учитывает сложность вопроса
 - b. всего теста
12. Информация о результатах тестирования должна сохраняться в текстовых файлах. Формат сохраняемого файла должен быть разработан и описан в пояснительной записке.
13. В обучающем режиме система по запросу пользователя должна предоставлять подсказки об ответах на тест.

График выполнения – санкции за несвоевременную сдачу этапа.

Дополнительные плюсы в карму

0. Шифрование файлов. Система должна предоставлять пользователю возможность зашифровать и расшифровать файл с вопросами (допускается реализовать консольный вариант).

1. Инсталлятор для Windows
2. Работа под Windows XP
3. Разнообразие типов вопросов, количество вопросов (более 50)
4. настройки тестирования, генерация тестовых заданий, режимы работы (см выше).
5. Демонстрация расширения теоретической подсистемы
6. Демонстрация расширения тестирующей подсистемы
7. Работа под Alt Linux 9.1
8. Инсталлятор для Linux