

Техническое задание на разработку программы "Класс объектов CRuler"

1. Введение	2
1.1 Формулировка задания заказчика.....	2
1.2. Наименование программы.....	2
1.3. Назначение и область применения	2
2. Требования к программе	2
2.1. Требования к функциональным характеристикам.....	2
2.2. Требования к надежности	3
2.2.1. Отказы из-за некорректных действий программиста	3
2.2.2 Спецификация сообщений об исключительных ситуациях.....	3
2.2.3. Отказы из-за некорректных действий пользователя	3
3. Условия эксплуатации	4
3.1 Требования к информационной и программной совместимости	4
3.2 Требования к исходным кодам и языкам программирования.....	4
4. Требования к программной документации.....	4
4.1. Предварительный состав программной документации.....	4
5. Стадии и этапы разработки	4
5.1. Стадии разработки	4
5.2. Этапы разработки.....	5
5.3. Содержание работ по этапам.....	5
6. Порядок контроля и приемки	5
6.1. Виды испытаний	5
6.2. Общие требования к приемке работы	Ошибка! Закладка не определена.

1. Введение

1.1 Формулировка задания заказчика

Разработать класс объектов CRuler (линейка отображения, типа линеек в Microsoft Word) со следующими функциями:

- установка левой и правой границы отображения;
- установка и модификация «сетки» (или параметров табуляции)
- сохранение/восстановление состояния, извещение и отработка запросов главного окна (получение параметров и текущих установок)

1.2. Наименование программы

Наименование программы: "Класс объектов CRuler". Далее *программный продукт* или *библиотека*.

1.3. Назначение и область применения

Программный продукт, точнее *библиотека* предназначена для создания и модификации так называемых *линеек отображения*. Под созданием здесь подразумевается написание программистом-пользователем кода, с использованием *библиотеки*, создающего *линейку отображения*.

Линейка отображения (Ruler bar), в свою очередь, это элемент графического интерфейса текстового редактора, предназначенный для "быстрого" форматирования текста документа. Уточним предназначения:

- форматирование выравнивания документа
- отображение и установка табуляций
- отображение и установка отступов для абзацев

2. Требования к программе

2.1. Требования к функциональным характеристикам

Создаваемые *библиотекой* линейки отображения должны поддерживать следующие функции:

- установка левой и правой границы отображения
- установка и модификация "сетки" (или параметров табуляции)
- сохранение и восстановление состояния, извещение и отработка запросов главного окна, получение параметров и текущих установок

Предварительный демонстрационный пример:



2.2. Требования к надежности

В данном разделе следует внести некоторую ясность:

- *Библиотека* разрабатывается непосредственно для *программиста*, имеющего необходимость создать линейку отображения.
- *Библиотека* косвенным образом разрабатывается для *пользователя*, имеющего необходимость воспользоваться текстовым редактором, в котором в качестве инструмента создания линеек отображения будет использоваться *библиотека*.

2.2.1. Отказы из-за некорректных действий программиста

Некорректные действия программиста как-то: недостаточная инициализация, неправильный порядок инициализации, некорректные значения инициализации и т.д. Т.е., в терминах практического программирования - исключительные ситуации должны быть соответствующим образом обработаны, учитывая приведенную ниже спецификацию.

2.2.2 Спецификация сообщений об исключительных ситуациях

В случае обнаружения ошибки программиста будет сформировано сообщение об ошибке, содержащее:

- номер строки и название файла, где ошибка произошла
- общий тип ошибки - см. п. 2.2.1
- предложение об исправлении: что нужно проверить и/или дописать

Пример:

Initialization exception.

111, a.cpp.

“Check initialization parameters”

2.2.3. Отказы из-за некорректных действий пользователя

Отказы из-за некорректных действий пользователя и вообще отказы времени выполнения кода *библиотеки* не допускаются.

3. Условия эксплуатации

3.1 Требования к информационной и программной совместимости

Библиотека должна иметь возможность использования на компьютере под управлением операционной системы Windows XP и только. Также *библиотека* должна быть легко расширяема, в случае если разработчику понадобится создать пакет обновлений: модифицировать код для использования *библиотеки* в новых версиях операционной системы Windows и/или добавить функциональность.

Код *библиотеки* должен корректно компилироваться (то есть без ошибок и предупреждений) с помощью Visual C++ Compiler, со стандартными настройками (то есть не требовать дополнительной настройки компилятора «из коробки»), входящего в состав IDE Microsoft Visual Studio версии не ниже 2005.

Библиотека должна представлять собой *статическую*, то есть результат работы это .lib файл и сопутствующими заголовочными .h файлами.

3.2 Требования к исходным кодам и языкам программирования

Библиотека будет реализована на языках C/C++.

Для графического отображения будет использован интерфейс WinAPI.

Техническую документацию *библиотеки* следует выполнить с помощью утилиты Doxygen версии не ниже 1.6.3.

4. Требования к программной документации

4.1 Предварительный состав программной документации

Состав программной документации должен включать в себя:

- техническое задание
- программу и методики испытаний
- техническую документацию программы

5. Стадии и этапы разработки

5.1 Стадии разработки

Разработка должна быть проведена в две стадии:

1. Разработка и утверждение технического задания
2. Рабочее проектирование

5.2. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. Разработка программного продукта
2. Разработка технической документации
3. Испытания программного продукта

5.3. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи
2. определение и уточнение требований к техническим средствам
3. определение требований к программному продукту
4. определение стадий, этапов и сроков разработки класса и технической документации на него
5. согласование и утверждение технического задания

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение и методики испытаний
2. проведение приемо-сдаточных испытаний
3. корректировка программы и программной документации по результатам испытаний

6. Порядок контроля

Для испытания программного продукта следует реализовать простейший текстовый редактор (используя WinAPI и стандартные средства) и создать линейки прокрутки, используя *библиотеку*.

6.1. Виды испытаний

Испытания *разработчика* должны проходить в соответствии с функциональными требованиями:

1. Создать линейку отображения
2. Модифицировать отдельно левую, правую границы отображения
3. Изменить параметр табуляции

4. Создать консоль. Вывести текущие параметры и установки линейки (т.е. состояние *линейки* отображения). Сохранить их. Прodelать шаги 1-3. Загрузить сохраненные параметры. Убедится в правильности сохранения\загрузки.
5. Вывести в консоль информацию, подтверждающую корректное извещение и обработку запросов клиента линейкой. (Например, выводить некоторое сообщение, в случае появления запроса)
6. Проверка «крайних» случаев: установка максимального\минимального параметра табуляции, распаивание и сворачивание окна клиента, установка левой и правой границы отображения в максимальное\минимальное положения.
7. Проверить корректность сохранения и загрузки параметров после испытания 6.
8. Получить параметры линейки стандартным предоставленным *разработчиком* образом, а затем реализовать некоторый запрос из окна клиента, который так же будет запрашивать параметры. Сравнить результаты.
9. Проверка корректного удаления объекта: программный вызов деструктора и пользовательское закрытие окна клиента.

Следует так же предоставить *заказчику* возможность проведения не менее четырех своих испытаний.