

**ЗАДАНИЕ ДЕМОНСТРАЦИОННОГО ЭКЗАМЕНА**

|  |  |
| --- | --- |
| **Код комплекта оценочной документации** | КОД 1.1-2023-2025 |
| **Год действия задания** | 2023 |
| **Номер варианта задания** | 3 |

**ОПИСАНИЕ ЗАДАНИЯ**

**Описание модуля 2:** **«Разработка программного обеспечения»**

Вы можете выбрать любую среду разработки и язык программирования из доступных, но должны сделать это обдуманно, придерживаться при их использовании профессиональных стандартов.

Обязательным требованием является обеспечение ограниченного доступа к продукту, возможности ввода и хранения данных.

Программный продукт должен быть готовым решением. Пользователи не должны устанавливать или настраивать СУБД, вручную переносить хранящиеся данные и т.п. Учтите: компьютер при проверке будет сконфигурирован точно также, как и ваш перед началом знакомства с ним.

**При выполнении модуля 2 ставятся следующие цели:**

1. Разработка программного продукта.

**При выполнении данного модуля 2 ставятся следующие задачи:**

1. Выбрать технологический стек для реализации программного продукта.
2. Разработать объекты баз данных, заполнить таблицы тестовыми данными.
3. Разработать программный продукт на основании предоставленных функциональных требований.

**Описание модуля 3:** **«Стандарты разработки программного обеспечения»**

Модуль отражает общий профессионализм решения: обратная связь системы с пользователем, стабильная работа всех разработанных программ, стиль кода на протяжении разработки всей системы, организация файловой структуры проекта, соблюдение культуры кодирования, комментарии к коду, умение работать с системой контроля версий.

**При выполнении модуля 3 ставятся следующие цели:**

1. Разработка кода программного продукта в соответствии с отраслевыми стандартами.

**При выполнении модуля 3 ставятся следующие задачи:**

1. Реализовать обратную связь системы с пользователем.
2. Соблюдать культуру кодирования.
3. Результаты работы предоставить в системе контроля версий.

*Название приложения*

Используйте соответствующие названия для ваших приложений и файлов. Так, например, наименование настольного приложения должно обязательно включать название компании- заказчика.

*Файловая структура*

Файловая структура проекта должна отражать логику, заложенную в приложение. Например, все формы содержатся в одной директории, пользовательские визуальные компоненты – в другой, классы сущностей – в третьей.

*Структура проекта*

Каждая сущность должна быть представлена в программе как минимум одним отдельным классом. Классы должны быть небольшими, понятными и выполнять одну единственную функцию (Single responsibility principle).

Для работы с разными сущностями используйте разные формы, где это уместно.

*Макет и технические характеристики*

Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

* разметка и дизайн (предпочтение отдается масштабируемой компоновке;
* должно присутствовать ограничение на минимальный размер окна;
* должна присутствовать возможность изменения размеров окна, где это необходимо;
* увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);
* группировка элементов (в логические категории);
* использование соответствующих элементов управления (например, выпадающих списков для отображения подстановочных значений из базы данных);
* расположение и выравнивание элементов (метки, поля для ввода и т.д.);
* последовательный переход фокуса по элементам интерфейса (по нажатию клавиши TAB);
* общая компоновка логична, понятна и проста в использовании;
* последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например, с помощью кнопки «Назад»);
* соответствующий заголовок на каждом окне приложения (не должно быть значений по умолчанию типа MainWindow, Form1 и тп).

*Обратная связь с пользователем*

Уведомляйте пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях, запрашивайте подтверждение перед удалением, предупреждайте о неотвратимых операциях, информируйте об отсутствии результатов поиска и т.п. Окна сообщений соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления. Также можно использовать визуальные подсказки для пользователя при вводе данных.

*Обработка ошибок*

Не позволяйте пользователю вводить некорректные значения в текстовые поля сущностей. Например, в случае несоответствия типа данных или размера поля введенному значению. Оповестите пользователя о совершенной им ошибке.

При возникновении непредвиденной ошибки приложение не должно аварийно завершать работу.

*Оформление кода*

Идентификаторы переменных, методов и классов должны отражать суть и/или цель их использования, в том числе и наименования элементов управления (например, не должно быть значений по умолчанию типа Form1, button3).

Идентификаторы должны соответствовать соглашению об именовании (Code Convention) и стилю CamelCase (для C# и Java) и snake\_case (для Python).

Допустимо использование не более одной команды в строке.

*Комментарии*

Используйте комментарии для пояснения неочевидных фрагментов кода. Запрещено комментирование кода. Хороший код воспринимается как обычный текст. Не используйте комментарии для пояснения очевидных действий. Комментарии должны присутствовать только в местах, которые требуют дополнительного пояснения.

Используйте тип комментариев, который в дальнейшем позволит сгенерировать XML- документацию, с соответствующими тегами (например, param, return(s), summary и др.)

**Требования к оформлению письменных материалов**

Все письменные материалы в ходе выполнения экзаменационного задания должны быть оформлены аккуратно, руководствуясь ГОСТ для оформления документов при необходимости.

**Представление результатов работы**

Все практические результаты должны быть переданы заказчику путем загрузки файлов на предоставленный вам репозиторий системы контроля версий git. Практическими результатами являются:

* исходный код приложения (в виде коммита текущей версии проекта, но не архивом),
* исполняемые файлы,
* прочие текстовые файлы.

Результаты работы каждой сессии должны быть загружены в отдельный репозиторий с названием «Сессия X» (X – номер сессии).

Для оценки работы будет учитываться только содержимое репозитория. При оценке рассматриваются заметки только в электронном виде (readme.md). Рукописные примечания не будут использоваться для оценки.

**Необходимые приложения**

**Необходимые приложения смотреть в папке «КОД 1.1 Приложения к вариантам».**

Приложение 1.Описание предметной области.pdf

Приложение 2. Сессия 1.pdf