

«Грабли» ВКР . Распространенные ошибки содержания и оформления ВКР

«Кто писал не знаю, а я дурак читаю»
А.П.Чехов. «Жалобная книга».

В пояснительной записке, как и в человеке все должно быть прекрасно. Перечисленные ниже «косяки» встречаются практически в каждой работе, несмотря на уровень и качество ее содержания. За бортом оставлена явная безграмотность, неспособность сформулировать сложные технические положения и явное пренебрежение к эстетике того, чем занимаешься.

Общие рекомендации

Пояснительная записка – это, прежде всего, текст. А не набор фраз, скриншотов, фрагментов кода, скриптов.

Стили. Перед началом заливки текста создайте стили, как минимум, для заголовков, перечислений, абзацев с отступом и без, названий рисунков, неформатируемого текста (цитаты кода, скриптов, настроек и пр). Тогда форматирование не будет «съезжать»

Список источников. На источники следует ссылаться при описании фреймворков, библиотек, лайфхаков, прототипов, аналогов, решений и пр. (официальные сайты, страницы загрузки, wiki, репозитории проектов).

Полнота изложения. Пояснительной записка должна содержать описание выполненной работы во всех аспектах (работа *квалификационная*), а не описание результата и внешнего вида:

- бизнес-анализ, обзор аналогов и сравнительный анализ;
- требования пользователей - функционал;
- инструменты и средства разработки (особенности, важные для этой работы);
- системная аналитика - предметная область, жизненный цикл сущностей, поведение, требования;
- графический интерфейс - макет, аспекты (навигация, производительность, информативность);
- архитектура - ключевые моменты, структура, паттерны, сервисы, форматы, протоколы, интерфейсы, компоненты, слои и их наполнение – бизнес-слой, БД;
- реализация - структура кода, структуры данных, алгоритмы;
- тестирование, развертывание и установка;
- внешний вид - руководство пользователя;
- цитаты артефактов в приложениях - исходный код, скрипты настройки, документы предметной области, API.

Список приведен для разработок, связанных с программированием приложений. Для других тематик он должен быть скорректирован в соответствии с предметной областью.

Структура документа, форматирование

Для автора очевидно. Автор знает некоторые ключевые моменты своей разработки и считает их само собой разумеющимися, поэтому не упоминает о них. Пример: проектируемый сервис дистрибуции музыки предполагает наполнение контента администратором (треки, информация об альбомах, исполнителях). Дальнейшая фабула;

- необходимая «админка» в проекте не разрабатывается;
- сама бизнес-модель в таком виде вызывает вопросы;
- автор не понимает вопроса «откуда берется контент (наполнение) сервиса».

Информативность на единицу площади. Эффективно используйте площадь листа не потому, что «в стране с бумагой напряженка», а потому, что малоинформативно и читать неудобно:

- межстрочный интервал текста в таблицах;
- пропорции столбцов в таблицах;

			Продолжение таблицы 6
/users/me/upload-image/	POST	Тело запроса (form-data): Прикрепленное изображение.	Обновление изображения пользователя.
/users/me/artists/	GET	-	Получение поправившихся исполнителей (своих).
/users/me/songs/	GET	-	Получение поправившихся треков (своих).
/users/me/albums/	GET	-	Получение поправившихся альбомов (своих).

Оформление перечислений: перечисление – часть связного текста, должно иметь заголовок. Не должно быть больше одной страницы (вариант – перечислить тезисно, потом каждый тезис отдельно раскрыть), избегать без необходимости 2-3уровневые.

[illegible]

Заголовок пункта/подпункта не может быть заголовком перечисления. В перечислениях не следует размещать рисунки, таблицы и т.п..

Орфография, синтаксис

Согласование членов предложения (синтаксис): «кем-чем», «о ком - о чем», склонение, спряжение, «...тся - ...тся».

При большой загрузке

...После выполнения запроса и получения данных **состояние экрана меняется** на «Content», при этом **упаковывая** в это состояние загруженные данные. (Состояние упаковывает ???).

Согласование членов предложения (семантика): члены предложения согласованы, но не с теми сущностями, о которых думал автор. Из-за этого возникает странная смысловая интерпретация текста, не заметная, кстати, при беглом прочтении.

Цитируя классиков: «Подъезжая к сией станции и глядя на природу в окно, у меня слетела шляпа. И. Ярмонкин». А.П.Чехов. «Жалобная книга»

Когда установлено время начала экзамена, он может быть начат по своему желанию

Запятая. Запятая – элемент синтаксиса, а не знак паузы или выделения *важного для автора* сочетания или слова. Ставятся: перечисления, сложносочиненные и сложноподчиненные предложения. Выделяются: причастные (после основного слова) и деепричастные (всегда), *вводные слова* (например, кроме того... - явно перечислены в языке).

Типичные ошибки: пропуск «закрывающей запятой» в обороте или придаточном предложении, лишние запятые – якобы *вводные слова*.

Для корректной работы приложения, проводилось модульное тестирование...

Включение в текст формальных спецификаций. В связном тексте следует использовать только имена (идентификаторы) компонент, а не их полные спецификации.

По нажатию на кнопку «отправить сообщение» выполняется запрос с отправкой сообщения на сервер «POST /answers/answerId/message», с помощью метода `performRequest(MessagingResult.self, route: StudentAPIRoute.getMessages(token), completion: @escaping (Result<MessagingResult, Error>) -> Void)`. Сообщения от собеседника загружаются через определенный промежуток времени без каких-либо действий от пользователя с помощью запроса «GET /answers/answerId/message» через метод `func performRequest(MessageList.self, route: StudentAPIRoute.sendMessage(token), completion: @escaping (Result<MessageList, Error>) -> Void)`

Стилистика

Старый серый ослик Иа-Иа стоял ... и думал о Серьёзных Вещах. Иногда он грустно думал: “Почему?”, а иногда: “По какой причине?”, а иногда он думал даже так: “Какой же отсюда следует вывод?”
«Винни Пух и все-все-все»

Псевдо-строгость изложения. Автору кажется, что надо употреблять побольше «научных слов» и оборотов типа «Как было показано выше», «из этого следует».

Слова-паразиты. Слова «определенные», «данные», «некоторые» ничего не добавляют, а только создают впечатление широкого подхода к теме.

...**Данное** приложение должно содержать **определенный** перечень требований:

Тем более, что слово *данные* имеет свой смысл в IT: ... в *данном* приложении *данные* загружаются из базы *данных*, «данной нам в ощущениях»

...Данная среда разрабатывается и поставляется Google и является рекомендованной средой для разработки под данную...

Схоластика. «Цели» и «задачи» - это хорошо, но не надо о них напоминать на каждой странице. Абстрактные, ни к чему не обязывающие рассуждения – сюда же.

Не считайте, что читающий страдает амнезией. Не надо постоянно напоминать о том, что было сказано выше, об общих положениях разработки и т.п..

Технологический жаргон. Если термин не является общеупотребительным и имеет распространенный эквивалент, то его не нужно использовать. Если же он используется, надо писать его в оригинале, а не на кириллице (*Activity* - *активити*).

...написание приложений возможно на девайсах фирмы Apple,

Ссылка в тему «Т.Толстая. Надежда и опора»: <https://www.litmir.me/br/?b=42954&p=1>

Тем более, не следует уподобляться слэнгу хиппи или Брайтон-Бич: «Вайтовые труйзера на зипперах с файновым лейблом не лефтором покете не нужны?»

Банальности для технического специалиста. Не следует долго объяснять, что такое «клиент-сервер».

Явный машинный перевод, образы и метафоры: часто встречается при копипасте материала. Надо следить за уровнем изложения, не скатываться в примитив, подбирать общепринятые термины в описываемой области.

...Приемщик принимает новые подключения и переданные данные и складывает в очередь, имея возможность сразу перейти к новым событиям. Опрашиватель выбирает из очереди произошедшие события и, если запрос клиента полностью пришел, передает его работнику.

Заезженные слова. Слова с расширенным толкованием и частым употреблением:

- «информация» в IT-технологиях – программисты работают с данными;
- система.

Стиль провинциального журналистики: витиеватость, многословие, сложность там, где можно сказать просто.

Сочинение на тему «Как я писал ВКР». Пояснительная записка – это не сочинение на тему «Как я писал ВКР». Это документ, описывающий разработку. Типичные ошибки:

- личные мотивы, вообще изложение от первого лица (Я использовал паттерн...). Изложение должно быть в безличностной форме – используется, использовано, состоит из...;
- использование будущего и прошедшего времени (в начале работы – БУДЕТ, в конце – БЫЛО). Исключение: результаты тестирования, исправление ошибок

Поэтому я решил изучить как работают такие приложения и попробовать создать свою реализацию в виде веб-приложения. Такая задача интересна тем, что она требует функционала, отличающегося от функционала традиционных веб-страниц.

Разработка приложения будет происходить нативным способом.

Для разработки будет использоваться язык программирования Swift

Для работы клиента студента будут использоваться следующие запросы

Так должно быть:

```
public class LEP500NNPanel extends LEP500BasePanel {
    public final int numClasses = Values.EState2Count;
    public static final int numOutput = Values.EState2Count;
    public final static String tmpCsv="temp.csv";
    //-----
    private int batchSize = 0;
    private int numHiddenLayers = 5;
    private int nEpoch = 1000;
    private ArrayList<MeasureFile> measureFilesU = new ArrayList<>();
    private ArrayList<MeasureFile> measureFilesS = new ArrayList<>();
    private ArrayList<MFSelection> selections = new ArrayList<>();
    private ArrayList<LEP500Params> params = new ArrayList<>();
    private ArrayList<AnalyseResult> results = new ArrayList<>();
    private EntityList<User> users = new EntityList<>();
    private HashMap<Integer, ConstValue> state2Map = new HashMap<>();
    private ArrayList<DLConfiguration> configs = new ArrayList<>();
}
```

