2-ой промежуточный отчет

Требования ко 2-му промежуточному отчету

1. Презентация
2. Распечатанное ТЗ на программу (ГОСТ 19.201-78)
3. Прототип программы (компилируемый программный код)

- Руководитель должен оценить состояние дел, исходя из 6-ти баллов
- Презентация и ТЗ должны быть подписаны руководителем
- Студенты, не получившие оценку руководителя, к отчету не допускаются

Требования к презентации

Слайд 1. Название работы, цель, требования (формулирует руководитель)

Слайд 2. Задачи работы(формулирует студент)

Слайд 3. Аннотация (Text abstract)

Слайд 4. Графическая аннотация (Graphical abstract)

Слайд 5. Содержание ПЗ (Названия разделов и два уровня вложения)

Слайды 6-7. Проект программы. Описание процесса проектирования в нотации UML (Use Case, Class diagram, остальные по готовности)

Слайды 8-9. Экранные формы прототипа

Слайды 5-6. Если в работу включена разработка информационного, математического и/или лингвистического обеспечения (ИО, МО и/или ЛО), графические решения и т.д., описать их в соответствии с требованиями: (структура БД, описание моделей и алгоритмов и т.д.).

Слайд 3. Аннотация

Аннотация к ВКР–краткое описание работы.

Размер аннотации – до половины листа А4.

Содержание аннотации должно отражать тему диплома, его объем, количество таблиц и рисунков.

Аннотация не содержит результатов работы, она описывает цели и показывает, что включено в эту работу.

Слайд 3. Аннотация. Плохой пример

В данном документе описываются проблемы автоматизации рабочего процесса в компании «Смарт Технолоджис», рассматриваются пути их решения, а также проектные решения, принятые в ходе разработки: архитектура системы, функциональная структура, пользовательский интерфейс. А дальше?

Слайд 4. Graphical Abstracts

- A Graphical Abstract is a single, concise, pictorial and visual summary of the main findings of the article.
- This could either be the concluding figure from the article or a figure that is specially designed for the purpose, which captures the content of the article for readers at a single glance.

Графический абстракт

- Графический абстракт не перегружен текстом, кроме картинок и функциональных связей между ними почти ничего не содержит (http://www.elsevier.com/graphicalabstracts).
- Графический постер обращен, главным образом, к правому полушарию и предусматривает одномоментное (симультанное) восприятие информации.
- Методы:
  - метод ментальных карт;
  - Метод Highlights» (http://www.elsevier.com/journalauthors/highlights -) содержание всей статьи в 2-5 коротких предложениях или звуковых файлов

Примеры графических абстрактов

Selected items [Add/Remove data]

Geography = Canada
Sex = Both sexes
Age group = 15 to 24 years

Labour force characteristics	Students	2016				
		July	August	September	October	November
Unemployment rate (rate) [8]	Total, all students and non-students	..	..	13.4	12.3	12.0
	Students	..	..	14.7	12.7	12.1
	Full-time students [11]	..	..	14.6	12.4	12.0
	Part-time students [12]	..	..	16.1	15.3	12.9
	Non-students [13]	..	..	12.3	11.9	12.0
Employment rate (rate) [10]	Total, all students and non-students	..	..	53.7	54.7	54.1
	Students	..	..	38.3	40.3	40.5
	Full-time students [11]	..	..	36.6	38.5	38.8
	Part-time students [12]	..	..	65.7	65.3	65.1
	Non-students [13]	..	..	76.8	78.0	77.9

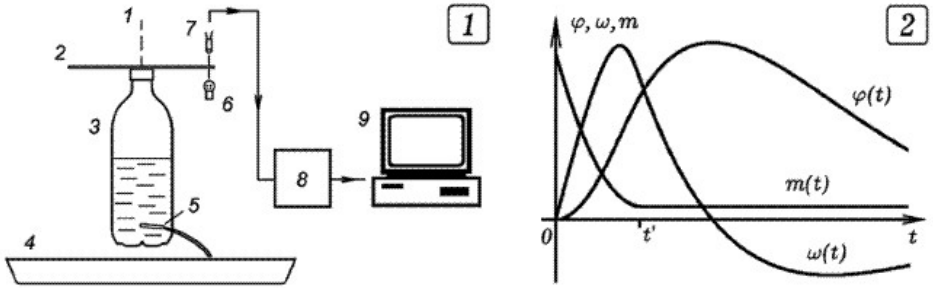
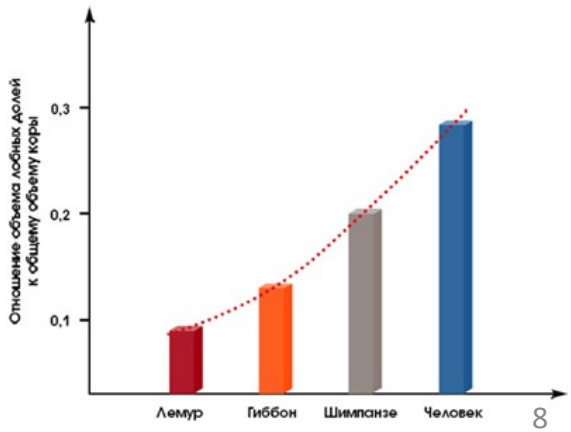
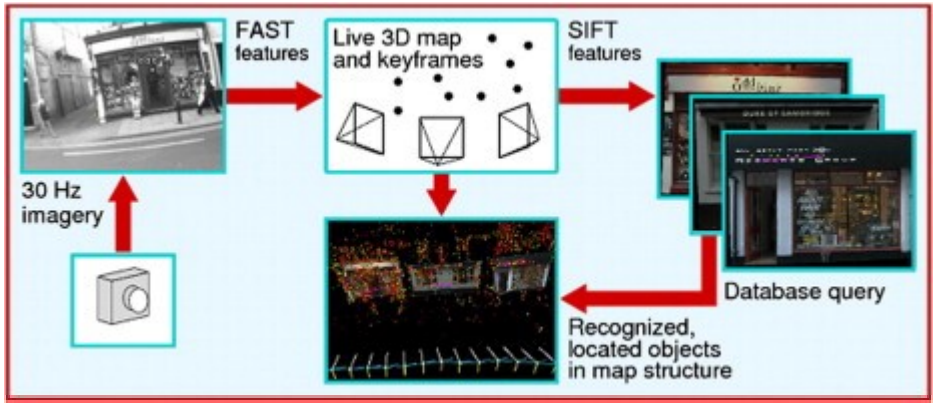
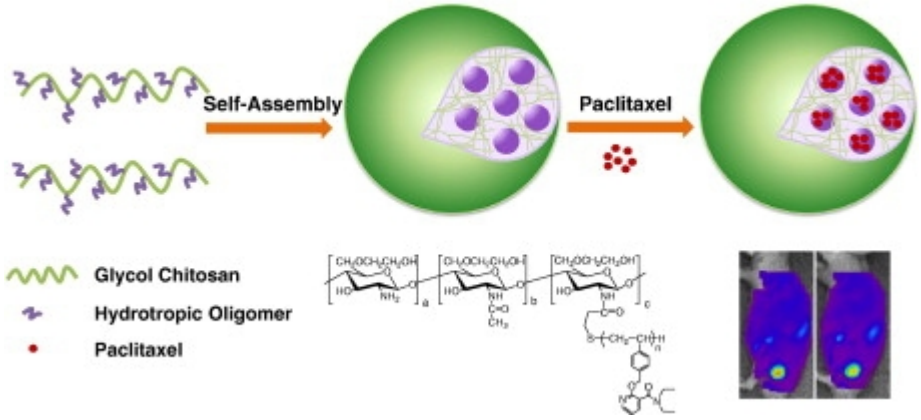


Рис. 1. Установка для изучения вращения колеса Сегнера
и результаты компьютерного моделирования.



Примеры графических абстрактов





Слайд 5. Содержание ПЗ. Плохой пример

Содержание ПЗ. Плохой пример

Слайды 6-7. Проект программы. Use-case диаграмма

- Use Case — это письменное описание того, как пользователь может взаимодействовать с системой, чтобы достичь определённой цели.
- Use-case диаграмма описывает варианты использования - процессы, которые могут выполнять пользователи
- Каждый Use Case представляет собой последовательность простых шагов, которые пользователь должен пройти, чтобы достичь цели.
- Use Case описывает, что делает система, а не как
- Use-case должна выражать требования к системе, а не детали ее реализации
- На Use Case изображаются:
  - Акторы (actor) – группы лиц или систем, взаимодействующих с нашей системой;
  - варианты использования (прецеденты) – сервисы, которые система предоставляет акторам;
  - комментарии;
  - отношения между элементами диаграммы.

Отношения между элементами Use Case. Отношение включения (*include*)

Отношение включения (*include*) указывает на то, что поведение одного прецедента включается в некоторой точке в другой прецедент в качестве составного компонента.

Включаемый прецедент должен быть обязательным для дополняемого (включение должно быть безусловным, а дополняемый вариант использования без включения не сможет выполняться), т.е. это отношение задает очень сильную связь.

Отношения между элементами Use Case. Отношение расширения (*extend*)

Отношение расширения (*extend*) отражает возможное присоединение одного варианта использования к другому в некоторой точке (точке расширения).

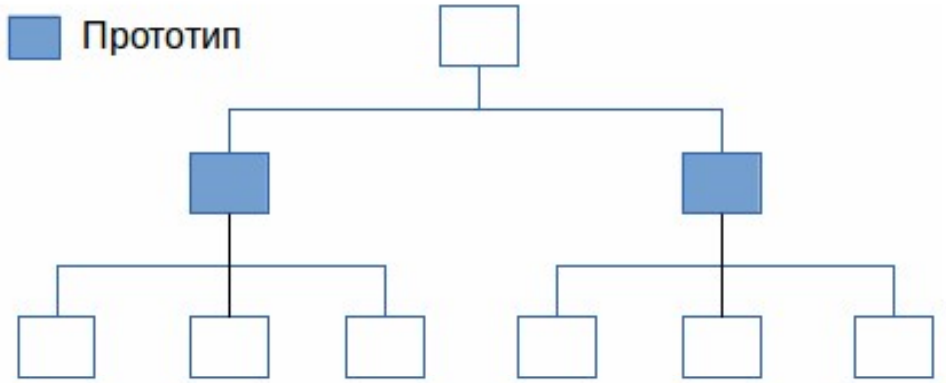
Расширяющий вариант использования выполняется лишь при определенных условиях и не является обязательным для выполнения основного прецедента.

Отношение расширения изображается стрелкой, направленной к расширяемому прецеденту, в отдельном разделе которого может быть описана точка расширения.

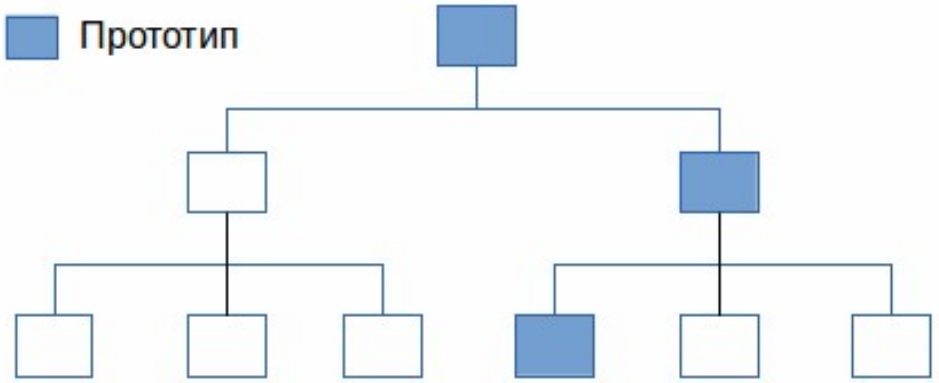
Условия расширения могут быть приведены в комментарии с ключевым словом *Condition*.

Расширение позволяет моделировать необязательное поведение системы, которое является условным и не изменяет поведение основного прецедента.

Прототипирование интерфейса. Виды прототипов



Горизонтальный прототип



Вертикальный прототип

Разделы ТЗ

ГОСТ 19.201-78 Техническое задание, требования к содержанию и оформлению)

Разделы ТЗ:

- введение;
- основания для разработки;
- назначение разработки;
- требования к программе или программному изделию;
- требования к программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки;

В ТЗ допускается включать приложения.

Раздел "Требования к программе или программному изделию"

2.4. Раздел "Требования к программе или программному изделию" должен содержать следующие подразделы:

- **требования к функциональным характеристикам**;
- требования к надежности;
- условия эксплуатации;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- требования к маркировке и упаковке;
- требования к транспортированию и хранению;
- специальные требования.

2.4.1. В подразделе "Требования к функциональным характеристикам" должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т. п.

Пример*.
Требования к программе
Требования к функциональным характеристикам

Состав выполняемых функций

Главные функции:

создание сюжетных линий;

контроль выполнения заданий сюжетной линии в процессе игры.

Основные функции:

- создание новой сюжетной линии;
- редактирование сюжетной линии;
- сохранение сюжетной линии в файл;
- загрузка сюжетной линии из файла;
- загрузка Lua-сценариев сюжетной линии;
- проверка заданий сюжетной линии;
- сохранение прогресса в файл;
- загрузка прогресса из файла.

Сервисные функции:

- создание нового задания;
- редактирование свойств задания;
- создание новой задачи;
- редактирование свойств задачи;
- создание нового решения;
- редактирование свойств решения;
- выбор Lua-сценария загрузки сюжетной линии;
- добавление переменных Lua для сохранения;
- удаление переменных Lua для сохранения;
- отмена операции;
- повтор отмененной операции;
- вызов справки.

*ВКР Алимова А., 2010.

Организация входных и выходных данных*

Входные данные

В процессе работы программы вводимой информацией являются параметры заданий, задач (идентификатор, название, описание, имена файлов Lua-сценариев), параметры решений (идентификатор и описание), связи между заданиями, задачами и решениями.

Входными данными являются Lua-сценарии, файлы с сюжетной линией, файлы сохранений.

Выходные данные

Выводимыми данными программы должна быть информация о сюжетной линии, ее элементах, предоставляемая игре, файлы сюжетной линии и файлы сохранений.

*ВКР Алимова А., 2010.

Требования к разработке ПО

Работа должна включать:

- исследование аналогов и прототипов (при их наличии) создаваемого ПО. Обоснование выбора представленных аналогов и прототипов, а также критериев их оценки;
- описание постановки задачи на проектирование с указанием основных функций, подлежащих проектированию, специфических требований к создаваемому ПО;
- представление архитектуры проектируемого ПО с графическим представлением основных компонентов ПО и связей между ними;
- описание процессов проектирования с использованием современных средств (например, диаграмм языка UML);
- представление иерархического описания подчинения функций создаваемого продукта;
- проектирование пользовательского интерфейса (если необходимо) – представление графического описания макетов экранных форм в следующей последовательности:
  - общие требования к дизайну ПО;
  - формы входной информации;
  - формы основных процессов или вычислений;
  - формы выходной информации.
- Эскизы форм, могут выполняться с применением специальных средств или от руки.

Требования к программному коду

Использование стандарта кодирования

- допускается использовать любой опубликованный стандарт кодирования, если рекомендуемый стандарт не соответствует специфике работы;
- для ПО, являющегося компонентой существующих систем, необходимо использовать стандарт кодирования, рекомендуемый разработчиками системы;
- комментарии к функциям должны содержать описание назначения, параметров и возвращаемого результата функции, даже если это не требуется используемым стандартом кодирования

Исходные файлы за авторством студента должны содержать в начале файла комментарий, содержащий:

- описание файла (1 абзац),
- ФИО студента (или псевдоним)
- знак копирайта (c),
- дату (год) создания файла, и его последнего изменения,
- контактную информацию.

Ставя подпись на документации к ВКР, студент утверждает свое авторство и берет ответственность за возможный плагиат.

Исходные коды программы должны сдаваться в электронном виде.

Файлы исходного кода, которые разработаны студентом, должны быть помещены в отдельную директорию с названием src, все сторонние библиотеки – в папку lib.

Требования к Web-приложениям

Разработка должна содержать ИО и МО (БД и алгоритмы, реализуемые на стороне сервера и/или клиента).

Сайт с использованием только статического HTML, или незначительно использующий JavaScript, не является WEB - приложением.

Описание ИО и МО должны соответствовать предъявляемым к ним требованиям.

Соблюдение стандартов кодирования

Выполнение проектирования структуры сайта и интерфейса (для не Web-сервисов) или проектирования API (для Web-сервисов)

Допускается использование CMS для разработки интерфейса Web-приложения.

Технологии реализации:

- ASP.NET,
- PHP,
- Perl,
- Java,
- JavaScript,
- другие технологии с их обоснованием выбора.

Требования к разработке ИО

- Описание процесса проектирования ИО (модель данных, ER-диаграммы, инфологическая модель, нормализация, типы данных).
- Обоснование выбора СУБД (если используется).
- Проектирование интерфейса.
- Обоснование выбора инструментария для разработки интерфейса (если есть).
- Разработка ИО (порядка 10-15 сущностей, если разработка ИО является основным компонентом), представление архитектуры, схемы данных, таблиц, связей и триггеров (если есть).
- Тестирование ИО:
- тестирование функционала,
- юзабилити,
- основные характеристики (целостность, непротиворечивость, минимальное дублирование, независимость, безопасность, для распределенных – синхронизация, транзакции).
- Разделение функций ИО (реализованных лично) и СУБД.

Требования к разработке МО.

Требования к представлению математических моделей

В случае представления в качестве результата работы математических моделей необходимо:

- определить цели моделирования;
- описать объект моделирования с точки зрения решаемой задачи;
- проанализировать способы формализации;
- сравнить подходы к решению задачи и обосновать выбор метода;
- разработать модели;
- оценить модели;
- проанализировать результаты моделирования.

Варианты реализации математических моделей:

- разработка собственного ПО (в соответствии с требованиями)
- использовать существующие инструментальные средства (необходимо обоснование выбора)

Примечание: целесообразно при формулировке темы ВРБ включать в постановку задачи разработку собственного ПО.

Требования к разработке МО (продолжение). Требования к разработке алгоритмов

- Алгоритмы должны быть описаны по шагам на естественном языке и/или с использованием псевдокода, блок-схем, R-графа.
- Алгоритмы должны быть верифицированы, то есть должны быть приведены результаты тестовых испытаний, которые показывают правильность работы алгоритмов.
- Должен быть дан общий анализ разрабатываемых алгоритмов:
  - точность расчёта и её зависимость от количества итераций;
  - класс алгоритма (его асимптотика),
  - зависимость времени работы алгоритма и объёма требуемой памяти от размера входных данных,
  - предлагаемые подходы к оптимизации (как алгоритмической, так и с точки зрения реализации),
  - способность к распараллеливанию (если это актуально).

Требования к разработке ЛО

Описание ЛО должно включать:

1. Описание грамматик разрабатываемых языков, с пояснениями к выбору нетерминалов, терминалов. Пример вывода произвольной наиболее полной цепочки языка, порожденного разрабатываемыми грамматиками. Приведение дерева вывода.

2. Алгоритм анализа цепочки на разрабатываемом языке или описание автомата разбора. Пример работы парсера для произвольной цепочки на разрабатываемом языке.

3. Описание алгоритма трансляции входной цепочки в выходную. Пример работы транслятора.

Грамматика языка должна быть записана в виде грамматики Хомского. Грамматика должна быть контекстно-свободной (приведенной к нормальной форме Хомского или Грейбаха) или автоматной.

Синтаксис языка должен быть описан в виде БНФ, РБНФ или диаграмм Вирта.

Для создания программы могут быть выбраны любые язык программирования и среда разработки, отвечающие всем требованиям разработчика.

Выбор языка программирования и среды разработки должны быть обоснованы.

Разрабатываемая программа должна реагировать на непредусмотренные и некорректные действия пользователя и выдавать соответствующие сообщения.

В программе должна быть предусмотрена контекстно-зависимая помощь.

Требования к разработкой графических и анимационных решений и их программной реализацией.

Требования к разработке геометрического обеспечения

В качестве графических решений могут выполняться:

- Геометрические 3D модели (создание или доработка):
  - единичные (например: детали, элементы сборки);
  - параметризованные;
  - сборочные;
  - изобразительные (например: дизайнерские решения, модели анимационных персонажей);
- 2D чертежи, схемы, шаблоны, карты, объекты.

Процесс разработки графических и анимационных решений должен быть описан в отдельном разделе пояснительной записки.

Требования к оформлению ВКР

Шифр ВКР

- В верхнем колонтитуле указывается идентификационный номер:
- Код института, кафедры
- XX Номер в приказе
- YY Год
- ZZ Код документа
- Вид работы–40 461 806–10.27–XX–YY.ZZ
-
-
- ZZ Код документа проставляется согласно ГОСТ 19.101-77.
- Распространенные коды программных документов:
  - техническое задание (ТЗ) – 90;
  - пояснительная записка (ПЗ) к техническому проекту (ТП) – 91;
  - ПЗ к рабочему проекту (РП) – 92;
  - руководство программиста – 33;
  - руководство оператора – 34;
  - руководство системного программиста – 32;
  - спецификация – 93;
  - описание программы – 13.

Виды программных документов и их коды (ГОСТ 19.101-77)

Код вида документа	Вид документа	Стадии разработки			
		Эскизный проект	Технический проект	Рабочий проект	
				компонент	комплекс
-	Спецификация	-	-	◐	●
05	Ведомость держателей подлинников	-	-	-	○
12	Текст программы	-	-	●	○
13	Описание программы	-	-	○	○
20	Ведомость эксплуатационных документов	-	-	○	○
30	Формуляр	-	-	○	○
31	Описание применения	-	-	○	○
32	Руководство системного программиста	-	-	○	○
33	Руководство программиста	-	-	○	○
34	Руководство оператора	-	-	○	○
35	Описание языка	-	-	○	○
46	Руководство по техническому обслуживанию	-	-	○	○
51	Программа и методика испытаний	-	-	○	○
81	Пояснительная записка	○	○	-	-
90-99	Прочие документы	○	○	○	○

● - документ обязательный;

◐ - документ обязательный для компонентов, имеющих самостоятельное применение;

○ - необходимость составления документа определяется на этапе разработки и утверждения ТЗ;

- - документ не составляют.