**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Практика разработки документации (на английском языке)

Practice of Documentation Development (in English)

**Язык(и) обучения**

английский

Трудоемкость в зачетных единицах: 4

Регистрационный номер рабочей программы: 050503

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Дисциплина «Практика разработки документации» является частью основной образовательной программы бакалавриата 09.03.04 «Программная инженерия» и предназначена для обучающихся, занимающихся написанием выпускной квалификационной работы (ВКР) по программной инженерии. В рамках данного дисциплины обучающиеся получают методическую помощь при написании текста ВКР. Такая помощь оказывается необходимой в виду объективной сложности данной задачи: в основном, у бакалавров совсем нет опыта написания более-менее сложных текстов. Кроме того, ВКР для них является новой и сложной задачей, и они испытывают значительные трудности в планировании и разработке всех составляющих этой работы в срок. В частности, они традиционно не уделяют должного внимания тексту ВКР. Следует также отметить, что в области программной инженерии как учебной дисциплины, в силу её молодости, отсутствуют общепринятые правила по написанию ВКР. Наконец, большие проблемы в написании документации существуют и в ИТ-индустрии, на что указывают практики и исследователи на протяжении последних 50-ти лет (Брукс, Парнас, Сомервилл и др.). А большинство ВКР по программной инженерии – это разработка и представление некоторой программной системы. Соответственно, и тексты ВКР по этой специальности оказываются, во многом, именно документацией к этим системам. Таким образом, в рамках данной дисциплины обучающиеся также получают опыт разработки документации к ПО.

Данная дисциплина является сугубо практической. Обучающимся предлагается осмыслить свою ВКР, которую они заканчивают в момент освоения данной дисциплины, спроектировать план текста ВКР и написать первые две версии текста по этому плану. Для проектирования предлагается использовать интеллект-карты (mind maps). Очень важно организовать проектирование текстов ВКР так, чтобы у обучающихся происходило структурирование их работы, а не формальное составление планов. Для этого важно идентифицировать ситуации, когда работы над ВКР не ведется, а в рамках данной дисциплины создаётся лишь некоторый выдуманный план. Бывает также, что работа над ВКР ведётся, но план все равно оказывается выдуманным. Последняя ситуация хорошо диагностируется при переходе от плана к тексту – план при этом не используется, или возникают значительные трудности с написанием текста. В этом случае необходимо снова перейти к интеллект-картам с учетом полученного урока. В целом же включение в рамках дисциплины требования по представлению текста ВКР (не окончательного), который бы соответствовал плану, оказывается решающим фактором для успешности деятельности обучающихся в рамках данной дисциплины. При этом текст не должен быть окончательным – иначе трудоёмкость проверки таких текстов для лектора данной дисциплины станет запредельной.

Цель изучения дисциплины: получение выпускниками помощи при написании текста ВКР, а также приобретение ими опыта в написания документации программного обеспечения и опыта проектирования при работе со сложной информацией.

Задачи курса: создание полноценного плана ВКР с помощью интеллект-карт, с вовлечением научного руководителя в деятельность по проектированию, а также создание двух версий текста ВКР, соответствующих этому плану.

The course is a part of the 09.03.04 bachelors educational program. It is intended for undergraduate students that are completing their bachelor thesis in Software Engineering. During the class, the students receive methodological help in writing their theses. This is necessary due to the objective complexity of the task. Generally, undergraduate students do not have any experience in writing complex texts. Besides, the bachelor thesis is a new and complicated task, and the students struggle with planning and developing thesis in time. Mostly, they do not pay close attention to the text development. It is important to note that the Software Engineering as an educational field, due to its young age, lacks established rules for writing theses. Finally, the software industry itself suffers from problems in documentation writing, which has been pointed out both by industrial developers and researchers during the last 50 years (F.Brooks, D.Parnas, I.Sommerville). At the same time, the majority of theses in Software Engineering represent the design and development of a software system. Thus, their papers are largely the documentation of the developed systems. Therefore, during this course students also acquire skills for writing software documentation.

This course is strictly practice-oriented. The students are supposed to re-think their theses, which they will be finishing during the course, devise a plan for its text and write the first two versions according to it. The designing process is facilitated using Mind Maps. It is very important to organize the design process in such a way that the students actually structurize their work without resorting to empty formalization. This requires identifying the situations where a student is not properly working on his thesis, and is just creating a made-up plan. There are also situations where a student is working on their thesis, but the plan he created during the course is still disconnected from reality. The latter is easily recognized during transitioning from the plan to the text: either the plan remains unused, or the student experiences significant trouble writing the paper. In this situation, the student needs to come back to the Mind Maps created during the class. In general, requiring that a student presents some version of their paper that corresponds to the devised plan is the key factor in making the course effective. At the same time, these texts should not be final, because otherwise reviewing all of them would become an insurmountable task for the lecturer.

The main goal of the Course is that students receiving assistance with their thesis papers, as well as acquiring experience in writing software documentation and designing projects with complicated information.

Course objective are in creating a detailed thesis plan using Mind Maps, involving the student's scientific advisor in the planning, as well as creating two versions of the thesis paper that correspond to the devised plan.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Программа курса предназначена для обучающихся 4 курса 8-го семестра, активно работающих над своими ВКР и заинтересованными в создании хороших текстов. Последнее очень важно – если обучающийся не заинтересован в собственном хорошем тексте, то работа по дисциплине быстро становится вырожденной, и нет методических методов сделать ее содержательной.

Кроме того, обучающиеся должны владеть средой LaTeX или Microsoft Word – одном из двух средств, в которых предлагается создавать тексты ВКР. Также обучающиеся должны владеть знаниями и навыками базовых курсов по программной инженерии, в частности, хорошо ориентироваться в следующих темах:

• процесс разработки ПО (в частности, разработка требований, проектирование, тестирование);   
• основы математической статистики и информационного поиска (постановка и проведение экспериментов);  
• проектировании ПО, включая знание языка UML.  
  
This course is intended for 4th year undergraduate students taking their 8th semester. They need to be actively working on their theses and motivated to write a good paper. The latter is highly important, because if a student has no interest in writing well, the work done during this course becomes irrelevant, and nothing else can provide it with meaning.  
Furthermore, the students need to be skilled in either LaTeX or Microsoft Word, because the two environments offered for thesis development. Basic knowledge and skills in Software Engineering are required, in particular:  
• software development process (specifically requirements, design, testing);  
• the basics of mathematical statistics and information retrieval;  
• software design including UML for creating good schemas of the software developed.

**1.3. Перечень результатов обучения (learning outcomes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование категории (группы) компетенций | Код и наименование компетенции | Планируемые результаты обучения, обеспечивающие формирование компетенции | Код индикатора и индикатор достижения универсальной компетенции |
|  | 1 | 2 | 3 | 4 |
| 1 | Общепрофессиональные компетенции | ОПК-1 – способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности; | Навыки по написанию документации программного обеспечения | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения |
| 2 | Общепрофессиональные компетенции | ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности | Навыки по написанию документации программного обеспечения | ОПК-2.2 Уметь проводить формализацию и алгоритмизацию поставленных задач |
| 3 | Профессиональные компетенции | ПКП-1 – способен проектировать программные системы; | Навыки по написанию документации программного обеспечения | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие |
| 4 | Профессиональные компетенции | ПКП-2 – способен использовать основные модели информационных технологий и способы их применения для решения задач в предметных областях; | Навыки по написанию документации программного обеспечения | ПКП-2.1 Уметь описывать алгоритмы компонентов, включая методы и схемы |
| 5 | Профессиональные компетенции | ПКП-3 – способен разрабатывать моделирующие алгоритмы и реализации их на базе языков и пакетов прикладных программ моделирования; | Проектирование при работе со сложной информацией | ПКП-3.1 Разработка модели бизнес-процессов заказчика |
| 6 | Профессиональные компетенции | ПКП-7 – способен систематизировать и применять знания о содержании основных этапов и тенденций развития программирования, математического обеспечения и информационных технологий; | Проектирование при работе со сложной информацией | ПКП-7.1 Умеет проводить технические и управленческие ревизии создаваемого программного средства |
| 7 | Универсальные компетенции | УК-1 – способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач; | Проектирование при работе со сложной информацией | УК 1.2. Определяет информацию, необходимую для решения поставленной задачи; |
| 8 | Универсальные компетенции | УК-4 – способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах); | Проектирование сложных текстов | УК-4.3. Ведет деловую переписку на русском языке с учетом особенностей стилистики официальных и неофициальных писем; |
| 9 | Универсальные компетенции | УКБ-3 - Способен понимать сущность и значение информации в развитии общества, использовать основные методы получения и работы с информацией с учетом современных технологий цифровой экономики, искусственного интеллекта и науки о данных, а также информационной безопасности | Понимание полезности текстов и документации в разных смыслах | УКБ-3.2. Точно определяет тип и форму необходимой информации. |

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Семинары – 10 ак.ч.  
Seminars: 10 academic hours

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 8 |  | 30 |  |  |  |  |  |  | 2 |  |  |  | 26 |  | 14 |  | 10 | 2 |
|  |  | 2-25 |  |  |  |  |  |  | 2-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО |  | 30 |  |  |  |  |  |  | 2 |  |  |  | 26 |  | 14 |  |  | 2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 8 |  |  | зачёт, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

**Основной курс Основная траектория Очная форма обучения**

Период обучения (модуль): Семестр 8

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| I. | Зачем нужен хороший текст ВКР | Семинары | 2 |
| Сам. работа по методическим материалам | 2 |
| II. | Управляющая структура дипломной записки | Семинары | 4 |
| Сам. работа по методическим материалам | 3 |
| III. | О проектировании текстов. Введение в интеллект-карты | Семинары | 4 |
| Сам. работа по методическим материалам | 2 |
| IV. | Постановка задачи и результаты ВКР – баланс с текстом | Семинары | 4 |
| Сам. работа по методическим материалам | 3 |
| V. | Как разделять описание архитектуры и реализации системы | Семинары | 2 |
| Сам. работа по методическим материалам | 2 |
| VI. | Применение UML и собирающая схема в тексте | Семинары | 2 |
| Сам. работа по методическим материалам | 2 |
| VII. | Как описывать требования к системе | Семинары | 2 |
| Сам. работа по методическим материалам | 2 |
| VIII. | Особенности написания введения | Семинары | 2 |
| Сам. работа по методическим материалам | 2 |
| IX. | Особенности написания обзора | Семинары | 2 |
| Самостоятельная работа по методическими материалам | 2 |
| X. | О встроенных в текст объектах | Семинары | 2 |
| Сам. работа по методическим материалам | 2 |
| XI. | Об экспериментах, тестировании и апробации, и использовании статистики | Семинары | 2 |
| Сам. работа по методическим материалам | 2 |
| XII. | Напоминание о русском языке | Семинары | 2 |
| Сам. работа по методическим материалам | 2 |
|  | Промежуточная аттестация | Самостоятельная работа | 14 |
|  |  | Зачет | 2 |

**Содержание модулей**

1. Зачем нужен хороший текст ВКР.

В рамках данного модуля происходит прояснение мотиваций обучающихся по написанию хороших текстов ВКР. Традиционно обучающиеся считают это неважным. Основная мотивация – чтобы то, что они сделали, было понятно другим людям (это достигается через текст). Также лектор объясняет обучающимся важность наличия качественной документации к программным системам и то, как данная тема связана с хорошими текстами их собственных ВКР. Наконец, выдвигается и объясняется тезис о научном характере текста ВКР (даже в том случае, если сама ВКР не является научной, но производственной). В рамках данного модуля обучающиеся получают точную информацию о критериях получения зачета по данной дисциплине, а также по режиму выполнения проверки и обсуждения результатов самостоятельной работы.

1. Управляющая структура дипломной записки.

Представляется шаблон оглавления текста ВКР по программной инженерии и две его редакции – для программной системы (промышленная ВКР) и для исследовательской (научной) ВКР. Излагается модель текста ВКР (триплет) результаты/постановка задачи/оглавление основной части. Основная идея этой модели – прослеживаемость (traceability) в тексте результатов.

1. О проектировании текстов. Введение в интеллект-карты.

В рамках данного модуля кратко излагаются основы работы с интеллект-картами. Также представляются программные инструменты по работе с интеллект-картами.

1. Постановка задачи и результаты ВКР – баланс с текстом.

Обсуждается подход по балансировке триплета результаты/постановка задачи/оглавление основной части, а также по его ориентации на конечный текст ВКР. Обращается внимание на трудности балансировки.

1. Как разделять описание архитектуры и реализации системы.

Обсуждается понятие архитектуры программной системы – основные решения (включая альтернативы и аргументацию, иногда – эксперименты и пилотные реализации альтернатив), структура всего приложения, описание функциональности. Более того, роль архитектуры системы в тексте ВКР – это концентрированное представление информации о системе в одном месте (иначе она оказывается размазанной по всему тексту, и читателю не знакомому с работой, трудно синтезировать своё единое понимание представленной работы). К реализации относим устройство отдельных модулей и основные алгоритмы. При этом код системы (классы, методы, переменные) не описываем.

1. Применение UML и собирающая схема в тексте.

Для описания архитектуры предлагается использовать UML диаграммы компонент и размещений, а также диаграммы классов и диаграмму взаимодействий. Первые три типа диаграмм предлагаются для описания статики системы, последняя – для динамики (т.е. для описания базового сценария работы системы). Т.о. предлагается, следуя UML, разделять описание статики и динамики системы. В рамках текста ВКР предлагается создать 1-3 несложных базовых диаграмм, на которых предложенная система представлена целиком. Эти диаграммы нужно корректно описать в тексте ВКР (объясняется, что это значит). Часто обучающиеся увлекаются созданием нетривиальных диаграмм и упускают вопрос связи их с текстом – диаграммы оказываются подвешанными в воздухе (другими словами – в пустоте).

1. Как описывать требования к системе.

Требования являются существенной частью описания инженерной системой. Если обучающийся участвовал в формировании требований (это не всегда выполяется, поскольку часто он получает готовые требования от своего научного руководителя), то важно их детализировать и описать. В рамках данного модуля обучающиеся выясняют на практике, что такое функциональные и нефункциональные требования, а также чем требования отличаются от архитектуры.

1. Особенности написания введения.

Обучающимся предлагается две модели введения – конуса и последовательно вложенных контекстов. Указывается на основную задачу введения – сообщить читателю всю информацию, необходимую для понимания постановки задачи. Обсуждается стиль введения – констатирующий, аргументирующий, но не описательный (не объяснительный). Вводится понятие контекста.

1. Особенности написания обзора.

Обсуждается один из основных элементов научного текста – отделение результатов (и их описаний), предложенных авторами, от уже существующих. Выделяется три вида информации, которая должна содержаться в обзоре: (i) используемые в ВКР технологии, (ii) related work (научная ВКР) или обзор сходных систем и решений, имеющихся на рынке (индустриальная ВКР), (iii) описание контекстного проекта ВКР.

1. О встроенных в текст объектах.

Встроенные объекты – это рисунки (схемы, диаграммы), таблицы, листинги кода и формулы. Основной тезис этого модуля – встроенные объекты должны помогать пониманию текста, а не текст должен объяснять эти объекты. Обсуждаются особенности создания встроенных объектов различных видов. Обсуждается реализация полноценной связи встроенных объектов с текстом ВКР.

1. Об экспериментах, тестировании и апробации, и использовании статистики.

Рассказывается о методе GQM (Goal Question Metrics). Акцентируется внимание на вопросах (questions) эксперимента и метриках для измерения его результатов. Обсуждаются методы статистики, способы и трудности их использования в качестве метрик. Обсуждается необходимость анализа экспериментальных данных, а также наличия конструктивных и интересных выводов.

1. Напоминание о русском языке

Обсуждается отличие разговорного языка от письменно. Обсуждается феномен программисткого жаргона (в случае русского языка обсуждаются также англицызмы, активно вторгающиеся из ИТ-мира, например, эмейл, смска). Предлагается избавляться от жаргона и разговорного языка в тексте. Также обсуждается казенный технический язык. Выдвигается гипотеза об его оживлении, предлагается освоить живой технический письменный язык. Обсуждаются правила оформления перечислений, правило тезис/расшифровка, различные правила по оформлению элементов текста (заголовков, подписей и ссылок и пр.).

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| I. | The necessity of a well-written thesis | Семинары | 2 |
| Сам. работа по методическим материалам | 6 |
| II. | The structure of the thesis | Семинары | 4 |
| Сам. работа по методическим материалам | 12 |
| III. | Designing papers. An introduction to Mind Maps | Семинары | 4 |
| Сам. работа по методическим материалам | 12 |
| IV. | Balancing research questions and thesis results with text | Семинары | 4 |
| Сам. работа по методическим материалам | 12 |
| V. | Separating the descriptions of software architecture and implementation | Семинары | 2 |
| Сам. работа по методическим материалам | 6 |
| VI. | Using UML. The basic diagrams of the thesis paper | Семинары | 2 |
| Сам. работа по методическим материалам | 6 |
| VII. | Describing software requirements | Семинары | 2 |
| Сам. работа по методическим материалам | 6 |
| VIII. | Writing an introduction | Семинары | 2 |
| Сам. работа по методическим материалам | 10 |
| IX. | Writing a literature review | Семинары | 2 |
| Самостоятельная работа по методическими материалам | 8 |
| X. | Text-embedded objects | Семинары | 2 |
| Сам. работа по методическим материалам | 6 |
| XI. | Experiments, testing, validation, and using statistics | Семинары | 2 |
| Сам. работа по методическим материалам | 8 |
| XII. | Reminding about Russian/English | Семинары | 2 |
| Сам. работа по методическим материалам | 6 |
|  | Assessment | Самостоятельная работа | 14 |
|  |  | Exam | 2 |

**Description of the content**

1. The necessity of a well-written thesis.

In this module, the motivation of writing the thesis paper well is discussed. The main motivation here is explaining the results of their work to other people through text clearly. Furthermore, the lecturer explains the importance of high-quality software documentation and how is it connected to their theses. Finally, the bachelor thesis paper is characterized as a scientific paper, even when it describes an industrial work. Here, the students are explained the precise criteria of passing the class as well as the process of reviewing and discussing their homework.

1. The control structure of the thesis.

The students are provided with two versions of a thesis paper template: for an industrial thesis and research thesis. The triplet model of the bachelor thesis is explained: research questions/results/list of contents. The main idea of this model is to provide traceability of results in the text.

1. Designing papers. An introduction to Mind Maps.

This module briefly explains the basics of working with Mind Maps and using Mind Map software.

1. Balancing research questions/thesis results with text.

An approach to balancing the research questions/thesis results/list of contents triplet with the text is discussed, as well as methods to tune it accordingly to the final text of the thesis. Special attention is paid to the obstacles encountered during the balancing process.

1. Separating the descriptions of architecture and implementation.

This module discusses the notion of software architecture: main software product decisions (including alternatives and argumentation, sometimes experiments and pilot implementations of alternatives), product structure (modules/components, classes, etc.), functionality description. Moreover, the role of architecture in the thesis paper is explained as to be a concentrated representation of all information on the system. Otherwise, it gets lost in the text, and the reader may struggle with understanding the work in its entirety. Description of software implementation, on the other hand, includes the structure of individual components/modules and specification of algorithms. The code of the system (classes, methods, variables), however, should not be described in thesis paper.

1. Using UML. The basic diagrams of the thesis paper

The students are supposed to use UML while describing the architecture of their software: component, class, deployment, and interaction diagrams. The first three categories of diagrams are static system’s views, and the last one is dynamic ones. Therefore, the students are supposed to separate the descriptions of the statics and the dynamics of the system. Further, the students are proposed to create 1-3 basic diagrams that represent the entire system missing insignificant details. They need to be adequately described in the text, and the way to do so is explained during this module. Often students get carried away creating non-trivial diagrams, which eventually become completely disconnected from the text itself.

1. Describing requirements.

Requirements are a significant part of the description of an developed software. If a student has participated in the development of the requirements (which is often not true, because students usually receive ready-to-use requirements from their supervisors), then it is important to specify and describe them. In this module, the students find out in practice what functional and non-functional requirements are, as well as what differentiates them from architecture.

1. Specifics of writing an introduction.

The main task of an introduction is pointed out, which is to inform the reader of everything needed to understand the problem formulation. The style of introductions is discussed – it has to be assertive and argumentative, but not descriptive or explanatory. The notion of context is introduced. This module discusses two models of an introduction: the conic model and the model of sequentially nested contexts.

1. Writing a literature review.

This module discusses one of the main components of a scientific paper – strict separating the new research results proposed by the authors from what has been already achieved in the area (shortly these new results are called as a paper contribution). There are three types of information that should be present in the literature review: (i) software technology used in the thesis, (ii) related work (for a scientific thesis) or a review of similar systems and solutions that already exist on the market (industrial thesis), (iii) the description of the thesis' context.

1. Text-embedded objects.

Text-embedded objects are figures (schemes and diagrams), table, code listings, and formulae.

The idea conveyed in this module is that these objects should be helpful for understanding text, not the other way around. The specifics of creating text-embedded objects of different types are discussed, as well as connecting them to text in a meaningful way.

1. Experiments, testing, validation, and using statistics.

This module describes the GQM (Goal/Question Method). Special attention is paid to experiment questions and metrics for measuring experimental results. Statistical methods are discussed, focusing on their applicability as metrics and the obstacles that could be encountered. Finally, this module explains the necessity of analyzing experimental data and presenting interesting and constructive results.

1. Reminding about Russian/English

The differences of speech and writing are discussed. The phenomenon of programmer slang is mentioned (concerning Russian, English words loaned from the software industry in particular). Students are encouraged to refrain from using slang and colloquialisms in their papers. This module also discusses the bureaucratese-like written language of technical documents, proposing to liven it up. In the end, different formatting rules are explained, such as the statement/explanation structure, rules for typesetting headlines, captions, references, enumerations, etc.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Лектору следует организовать лекционно-практический поток занятий, начиная с лекций и полностью смещаясь со второй половины курса в практическую сферу. На составление интеллект-карт может потребоваться от 2х до 6-ти итераций для каждого обучающегося, на написание первых версий текста ВКР – 2-3 итерации.   
Лектору также следует наладить рабочий процесс по высылке и проверке домашних заданий, в частности, следует назначить точное время высылки сделанных заданий, чтобы у преподавателя была возможность их проверить перед занятием, а также согласовать форматы для оформления заданий.  
  
Для освоения дисциплины обучающиеся должны активно заниматься своими ВКР и регулярно посещать данный курс, активно выполнять домашние задания и вовлечь научного руководителя в проектирования текста ВКР.   
В рамках данной дисциплины обучающийся должен представить полноценный план ВКР с помощью интеллект-карт, вовлечь в деятельность по проектированию научного руководителя, суметь объяснить данный план лектору, а также предоставить две версии текста ВКР, соответствующих этому плану. Вторая версия делается обучающимся основе комментариев лектора к первой; при этом финальная (вторая) в рамках дисциплины версия текста ВКР должна реализовать все пункты оглавления до такой степени, чтобы было понятно содержание этих пунктов и всей работы.   
  
The course is organized starting with lectures for the first half and moving on to practice for the second one. Developing Mind Maps may require from 2 to 6 iterations for each student, and writing first versions of their papers may take 2 or 3 iterations.  
The practical process baseline should be set up. It includes of sending in and reviewing homework for each student, in particular, by setting a specific deadline for each assignment, so that the lecture will have enough time to review them before the next class. Furthermore, the formatting of each assignment needs to be agreed on as well.  
To be successful in passing this course, the students need to actively work on their theses, attend the class, complete their homework assignments and involve their supervisors into paper design.  
During the course, students are required to present a detailed thesis plan using Mind Maps, involve their supervisors into the planning, explain the plan to the lecturer, and present two version of their thesis paper that correspond to their plan. The second version here is completed after the lecturer comments on the first one. At the same time, this final version prepared during the course should implement all points on the list on contents so detailed as to make each of them and the work itself understandable for the lecture.

**3.1.2 Методическое обеспечение самостоятельной работы**

Совершенно необходимо прослушать установочную лекционную часть курса, а также лично пообщаться с лектором. Самостоятельно дисциплина не может быть освоена.   
  
Attending the lecture part of the course is obligatory, as well as a personal consultation with the lecturer. The course can not be passed unassisted.

**3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

Текущий контроль успеваемости состоит в выполнении всех практических заданий по курсу, а также в сдаче теоретического зачета – т.е. хорошего ответа на 1 вопрос из списка вопросов, представленного ниже.

Earning a passing grade in this course consists of completing all assignments and passing an oral test – i.e. a good answer to one of the questions from the list below.

В случае нарушения любого из условий получения зачёта обучающийся аттестуется отрицательно. При выполнении условий оценивание осуществляется по критериям, представленным в таблице ниже.

Итоговый\_процент\_освоения\_курса =

= 60% \* доля\_выполненных\_практических\_заданий +

+ 40% \* полнота\_ответа\_на\_теоретический вопрос.

When any of above conditions are not met, the student is assessed negatively. When assessed positively, following criteria are used (see table below).

Final\_completeness\_percentage =

= 60% \* fraction\_of\_practical\_assignments\_completed +

+ 40% \* completeness\_of\_oral\_test.

|  |  |  |
| --- | --- | --- |
| **Итоговый процент освоения курса (Final\_completeness\_percentage), %** | **Оценка СПбГУ (SPbU mark)** | **Оценка ECTS (ECTS mark)** |
| 0-49 | Не зачтено  (not passed) | F |
| 50-60 | Зачтено  (passed) | E |
| 61-69 | D |
| 70-79 | C |
| 80-89 | B |
| 90-100 | A |

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

Контроль выполнения практических заданий осуществляется путем проверки на соответствие демонстрируемого обучающимся результата требованиям задания, перечисленным в п. 3.1.1.

Пример списка вопросов, выносимых на зачет.

1. Управляющая структура дипломной записки.
2. Введение в интеллект-карты.
3. Основные положения проектирования текстов.
4. Постановка задачи и результаты ВКР – баланс с текстом.
5. Как разделять описание архитектуры и реализации системы.
6. Применение UML и собирающая схема в тексте. Виды используемых UML-диаграмм.
7. Особенности написания введения.
8. Особенности написания обзора: обзор используемых в ВКР технологий.
9. Особенности написания обзора: related work (научная ВКР) или обзор сходных систем и решений, имеющихся на рынке (индустриальная ВКР)
10. Особенности написания обзора: описание контекстного проекта ВКР.
11. Функциональные не нефункциональные требования к ПО.
12. Особенности описания требований в тексте.
13. Виды встроенных в текст объектов. Общий принцип "работы" встроенных объектов.
14. Правила разработки рисунков.
15. Проектирование таблиц.
16. Разработка и оформление листингов кода. Псевдокод.
17. Правила оформления формул.
18. Правила оформления перечислений.
19. Написание текста в стиле тезис/объяснение.
20. Метод GQM.
21. Об экспериментах, тестировании и апробации,
22. Об использовании статистики в экспериментах.

The completion of assignments is controlled via checking if the achieved result corresponds to the aforementioned criteria (3.1.1)

An example question list for the oral test:

* 1. The control structure of the thesis.
  2. An introduction to mind maps.
  3. The guidelines of paper design.
  4. Problem formulation and thesis results: balancing with writing.
  5. Separating descriptions of architecture and implementation of a software system.
  6. Using UML and the aggregating diagram in text.
  7. Specifics of writing an introduction.
  8. Specifics of writing a review: reviewing the technologies used in the thesis.
  9. Specifics of writing a review: related work (scientific thesis) or a review of similar systems and solutions on the market (industrial thesis).
  10. Specifics of writing a review: thesis context.
  11. Functional and non-functional system requirements.
  12. Specifics of describing system requirements.
  13. Types of text-embedded objects. General idea of how they work.
  14. Rules of figure development.
  15. Table design.
  16. Developing and typesetting code listings. Pseudocode.
  17. Formulae typesetting rules.
  18. Enumeration typesetting rules.
  19. Writing text according to the "thesis/explanation" structure.
  20. The GQM method.
  21. Experiments, testing, validation.
  22. Using statistics for experiments.

|  |  |  |
| --- | --- | --- |
| № | Код индикатора и индикатор достижения универсальной компетенции | Контрольно-измерительные материалы (КИМ) (тестовые вопросы, контрольные задания, кейсы и пр.) |
|  | 1 | 2 |
| 1 | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения | ответ на вопрос теоретического зачёта и практическое задание из п.3.1.1 независимо оцениваются по шкале от 0 (не сделано/нет ответа) до 10 (очень хорошо), после чего усредняются. Результат переводится в диапазон от 0 до 100 |
| 2 | ОПК-2.2 Уметь проводить формализацию и алгоритмизацию поставленных задач | задание из п.3.1.1 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 3 | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие | ответ на вопрос теоретического зачёта оценивается по шкале от 0 (нет ответа) до 100 (очень хорошо). |
| 4 | ПКП-2.1 Уметь описывать алгоритмы компонентов, включая методы и схемы | задание из п.3.1.1 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 5 | ПКП-3.1 Разработка модели бизнес-процессов заказчика | ответ на вопрос теоретического зачёта оценивается по шкале от 0 (нет ответа) до 100 (очень хорошо). |
| 6 | ПКП-7.1 Умеет проводить технические и управленческие ревизии создаваемого программного средства | ответ на вопрос теоретического зачёта оценивается по шкале от 0 (нет ответа) до 100 (очень хорошо). |
| 7 | УК 1.2. Определяет информацию, необходимую для решения поставленной задачи; | задание из п.3.1.1 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 8 | УК-4.3. Ведет деловую переписку на русском языке с учетом особенностей стилистики официальных и неофициальных писем; | задание из п.3.1.1 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 9 | УКБ-3.2. Точно определяет тип и форму необходимой информации. | ответ на вопрос теоретического зачёта и практическое задание из п.3.1.1 независимо оцениваются по шкале от 0 (не сделано/нет ответа) до 10 (очень хорошо), после чего усредняются. Результат переводится в диапазон от 0 до 100 |

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.1.5 Methodological materials for students to assess the content and quality of the educational process**

The assessment of the content and quality of the educational process by the students in the discipline is implemented in the manner established at St. Petersburg State University.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К преподаванию дисциплины могут быть допущены преподаватели, имеющие диплом о высшем техническом образовании по направлениям «Программная инженерия», «Математическое обеспечение и администрирование информационных систем» и другим смежным направлениям, связанным с информационными технологиями.  
  
This course can be taught by instructors who have received a master degree in Software Engineering, Software and Administration of Information Systems, and other similar fields. It is also required to have an experience in software documentation and research papers writing.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Специальных требований нет.  
  
No special requirements.

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

Учебные аудитории для проведения учебных занятий, оснащенные стандартным оборудованием, используемым для обучения в СПбГУ в соответствии с требованиями материально-технического обеспечения.

No special requirements.

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Стандартное оборудование, используемое для обучения в СПбГУ. MS Windows, MS Office, Mozilla FireFox, Google Chrome, Acrobat Reader DC, WinZip, Антивирус Касперского

No special requirements.

**3.3.3 Характеристики специализированного оборудования**

Специализированное оборудование не требуется.  
No special requirements.

**3.3.4 Характеристики специализированного программного обеспечения**

Специализированное программного обеспечения не требуется.  
No special requirements.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Для аудиторий с маркерными досками необходимы стирающиеся маркеры в объеме, достаточном для проведения курса. Для аудиторий с меловыми досками необходим мел в объеме, достаточном для проведения курса.

**3.4. Информационное обеспечение**

**3.4.1 Список литературы**

1. Annesley, T., 2010. Clinical Chemistry, 56 (3), 56(4), 56(5), 56(6), 56(7). – ЭР открытого доступа в сети Интернет: http://clinchem.aaccjnls.org/collection/clinical-chemistry-guide-scientific-writing?page=1

**3.4.2 Иные информационные ресурсы**

https://writingcenter.gmu.edu/ – The Writing Center at George Mason University.  
https://www.vanderbilt.edu/writing/ – Vanderbilt University Writing Center.  
• Сайт Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/>

• Электронный каталог Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/cgi-bin/irbis64r/cgiirbis_64.exe?C21COM=F&I21DBN=IBIS&P21DBN=IBIS>

• Перечень электронных ресурсов, находящихся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/>

• Перечень ЭБС, на платформах которых представлены российские учебники, находящиеся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/browse?name=rures&resource%20type=8>

**Раздел 4. Разработчики программы**

Кознов Дмитрий Владимирович Доктор технических наук доцент Профессор кафедры системного программирования d.koznov@spbu.ru  
  
Dmitry Koznov Doctor of technical sciences Docent Профессор кафедры системного программирования d.koznov@spbu.ru