

Министерство науки и высшего образования Российской Федерации

Томский государственный университет
систем управления и радиоэлектроники

М. Е. Антипин

Программная инженерия

Методические указания по проведению практических занятий

Томск
2023

УДК 004.02
ББК 3стд2-02
А 72

Рецензент:

Лобода Ю.О., доцент каф. управления инновациями ТУСУР,
канд. пед. наук

Антипин, Михаил Евгеньевич

А 72 Программная инженерия: Методические указания по проведению практических занятий/ М.Е. Антипин. – Томск: Томск. гос. ун-т систем упр. и радиоэлектронники, 2022. – 10 с.

Методические указания содержат рекомендации и материалы, необходимые для проведения практических занятий по дисциплине «Программная инженерия». Для студентов высших учебных заведений.

Одобрено на заседании кафедры УИ, протокол № 1 от 31.08.2022.

УДК 004.02
ББК 3стд2-02

© Антипин М.Е., 2023
© Томск. гос. ун-т систем упр. и радиоэлектронники, 2022

Оглавление

1. Общие положения	4
2 Общие требования к проведению практических занятий	5
3 Техническое обеспечение практических занятий.....	6
4 Прием результатов выполнения практических заданий.....	6
5 Терминология дисциплины.....	7
6 План практических занятий.....	8
7 Практические задания	9
Список рекомендуемой литературы.....	10

1. Общие положения

Данные методические указания разработаны для студентов, обучающихся в Томском государственном университете систем управления и радиоэлектроники (далее - Университет).

Структура дисциплины «Программная инженерия» предполагает проведение практических занятий. Практические занятия предназначены для закрепления материала, полученного в лекционном курсе, самостоятельного изучения и обсуждения материалов дисциплины, предусмотренных рабочей программой. Полученные навыки и знания могут быть полезны при управлении проектами разработки программных продуктов, моделировании, оптимизации, реинжиниринге бизнес-процессов организации, специализирующейся в разработке программного обеспечения, а также при самостоятельной разработке программ. Рекомендации по выполнению самостоятельной работе студентов приведены в соответствующих методических указаниях.

В ходе проведения практических занятий студентам прививаются навыки поиска информации, работы с учебно-методической документацией, умения увязывать теоретические знания с практикой, четко излагать свои мысли, отвечать на вопросы, оформлять и представлять результаты работы.

Рекомендации подготовлены с целью помочь студентам в успешном освоении дисциплины и подготовке и прохождении промежуточных этапов аттестации.

2 Общие требования к проведению практических занятий

Практические занятия по дисциплине «Программная инженерия» проводятся согласно учебному расписанию отдельно для каждой группы студентов очной формы обучения. В ходе практических занятий студент участвует в обсуждении темы, обозначенной на предыдущем занятии и выполняет практические задания, полученные от преподавателя. Практические задания выполняются студентами очной формы обучения индивидуально под контролем со стороны преподавателя. Все консультации осуществляются преподавателем.

Во время проведения практических занятий студентам в аудитории запрещается:

- Разговаривать между собой на любые темы без разрешения преподавателя.
- Консультировать друг друга.
- Передавать друг другу материалы, являющиеся результатом выполнения заданий.
- Производить шум, мешающий остальным сосредоточиться на выполнении задания.
- Пользоваться наушниками, берушами и другими приспособлениями, не позволяющими отчетливо слышать указания преподавателя.
- Читать литературу, конспекты и другие записи, не относящиеся к изучаемому предмету.
- Находиться в помещении аудитории в верхней одежде, если температура выше 18°C.
- Приносить верхнюю одежду с собой и размещать ее на стуле/столе, если в учебном корпусе работает гардероб.

В случае однократного нарушения преподаватель должен предупредить студента. При повторном нарушении в течении одного занятия студент из аудитории удаляется.

Студент имеет право:

- Уточнять полученные задания у преподавателя.
- Пользоваться любыми доступными методическими материалами по данной дисциплине.
- Просить консультации у преподавателя, если он в текущий момент не распределяет задания, не принимает выполненные работы и не консультирует другого студента.
- Пользоваться для выполнения практических заданий собственным ноутбуком или планшетным компьютером.

Преподаватель, давая консультацию студенту, указывает раздел технической документации или методической литературы, в которой имеется ответ на вопрос студента. Если необходимые сведения в документации и литературе отсутствуют, то преподаватель должен дать устные пояснения или продемонстрировать практические действия, приводящие к требуемому результату для повторения студентом.

3 Техническое обеспечение практических занятий

Практические занятия должны проводиться в аудитории, оборудованной:

- Доской и маркерами.
- Проектором и экраном.
- Персональными компьютерами, не менее одного на двух студентов группы, удовлетворяющих требованиям:
 - доступ в сеть Internet;
 - современный графический редактор для разработки моделей и схем.
 - программный пакет для управления программными проектами (Scrum, Trello, Jira и т.п.);
 - пакет офисных приложений для разработки текстов проектных документов.

4 Прием результатов выполнения практических заданий

За выполнение каждого задания преподаватель выставляет студенту оценку. Оценка выполнения задания складывается из трех равнозначных компонентов:

- Время выполнения задания. Фиксируется с момента получения задания до момента сдачи отчета. Измеряется в астрономических часах. Сравнивается с нормативным временем выполнения.
- Полнота и правильность выполнения задания. Экспертная оценка преподавателя.
- Аккуратность при выполнении текстовых и графических материалов.

Во время приема выполненной работы преподаватель вправе требовать у студента обоснования представленных материалов.

Преподаватель должен объявить студенту поставленную ему оценку за выполнение задания, а в случае возникновения непонимания, объяснить причины ее выставления. В случае, если оценка неудовлетворительно, студент имеет право повторно предъявить результат выполнения, но не более двух раз в течение одного занятия. При этом для вычисления оценки время, затраченное на исправление, прибавляется к общему времени выполнения задания.

Выставленная оценка влияет на оценку студента по контрольной точке и среднюю оценку за практические занятия.

До конца семестра студент должен получить оценку по всем заданиям, предусмотренным настоящими указаниями. За работы, результаты выполнения которых не были предъявлены преподавателю для оценивания, выставляется оценка неудовлетворительно. Студенты, имеющие среднюю оценку за практические занятия ниже удовлетворительной, к итоговой аттестации по предмету не допускаются.

5 Терминология дисциплины

Чтобы свободно ориентироваться в материалах дисциплины студенту следует ознакомиться с применяемой терминологией:

- Программа (program) – это набор операторов, который может быть представлен как единое целое в некоторой вычислительной системе и который используется для управления поведением этой системы.
- Программирование (в узком смысле) – процесс кодирования и отладки программы в рамках реального проекта.
- Программирование (programming) (в широком смысле) – все технические операции, необходимые для создания программы, включая анализ требований и все стадии разработки и реализации.
- Программная инженерия - системный подход к анализу, проектированию, оценке, реализации, тестированию, обслуживанию и модернизации программного обеспечения. Программная инженерия занимается разработкой систематических моделей и надежных методов производства высококачественного программного обеспечения, и данный подход распространяется на все уровни — от теории и принципов до реальной практики создания программного обеспечения. ПИ является отраслью информатики (computer science), это инженерная дисциплина, которая изучает вопросы построения компьютерных программ, отражает закономерности развития программирования, обобщает опыт программирования в виде комплекса знаний и правил регламентации инженерной деятельности разработчиков ПО.
- Технология программирования – процессы программной инженерии, методы программирования, инструментальные средства.
- Программное обеспечение – совокупность программ для обработки информации и комплект документации, необходимой для ее эксплуатации.
- Жизненный цикл – развитие системы, продукта, услуги, проекта начиная со стадии идеи и заканчивая завершением применения.
- Конфигурационное управление (англ. Software configuration management, SCM) в программной инженерии — комплекс методов, направленных на систематический учёт изменений, вносимых разработчиками в программный продукт в процессе его разработки и сопровождения, сохранение целостности системы после изменений, предотвращение нежелательных и непредсказуемых эффектов, формализацию процесса внесения изменений.

6 План практических занятий

Практические занятия проводятся по четырем разделам дисциплины:

Раздел 1. Введение в предметную область.

Тема 1. Использование библиотек электронных документов для поиска стандартов по программной инженерии (ГОСТ).

Тема 2. Международные информационные системы для стандартов International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE), Military Standard System (MIL STD) в области программной инженерии.

Тема 3. Процессы жизненного цикла продукции.

Раздел 2. Жизненный цикл программных продуктов. Основные модели жизненного цикла.

Тема 1. Разработка требований.

Тема 2. Уровни требований.

Тема 3. Выявление требований.

Тема 4. Анализ требований.

Тема 5. Спецификация требований.

Тема 6. Проверка требований.

Тема 7. Управление требованиями.

Раздел 3. Формальные и полужформальные методы спецификации, верификации и доказательства правильности.

Тема 1. Формальные и полужформальные методы при разработке программных средств.

Тема 2. Диаграммы классов.

Тема 3. Диаграммы потоков данных.

Тема 4. Нотации Йордана и Гейна-Сарсон.

Тема 5. Диаграмма SADT.

Тема 6. Событийно-ориентированная диаграммы функций (EPC).

Тема 7. Нотации семейства IDEF.

Тема 8. Диаграмма вариантов использования.

Тема 9. Таблицы и деревья решений.

Тема 10. Причинно-следственные диаграммы.

Тема 11. Дерево отказов.

Тема 12. Язык UML.

Раздел 4. Теория и методы проектирования. Новые подходы к разработке.

Тема 1. Методология обеспечения качества в программной инженерии.

Тема 2. Методические документы и комплекс международных стандартов.

Раздел 4. Теория языков программирования. Классификация языков программирования.

Тема 1. Методы и технологии программирования.

Тема 2. Функциональное и логическое программирование.

Тема 3. Модульное программирование.

Тема 4. Структурное программирование.

Тема 5. Объектно-ориентированное программирование.

Тема 6. Экстремальное программирование.

Тема 7. Визуальное программирование

Тема 8. Методология обеспечения качества в программной инженерии.

7 Практические задания

1. Основные модели жизненного цикла программных продуктов и этапы жизненного цикла.
2. Методологии разработки программ и жизненные циклы для разных методологий.
3. Методы формальной спецификации требований.
4. Модели.
5. Методы доказательства корректности программ.
6. Формальные методы в тестировании.
7. Полуформальные методы разработки и тестирования.
8. Классификация языков программирования.
9. Синтаксис языковых конструкций, описания переменных, записей, структур и функций.
10. Работа с регулярными выражениями.
11. Конечных автоматы.
12. Структурные, процедурные, объектно-ориентированные. функциональные, мультипарадигмальные, графические, баз данных, аппаратные, параллельные языки.
13. Теория проектирования программных продуктов.
14. Процессы разработки программных продуктов.

Список рекомендуемой литературы

1. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 147 с.
2. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 432 с.
3. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2020. — 235 с.
4. Практическое введение в программную инженерию : учебное пособие для вузов / В. К. Волк. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 100 с.