

Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(государственный университет)»



«УТВЕРЖДАЮ

Проректор по учебной работе
и экономическому развитию

Д.А. Зубцов

Рабочая программа дисциплины (модуля)

по дисциплине: Программная инженерия
по направлению: Прикладные математика и физика (бакалавриат)
профиль подготовки: Компьютерные технологии и интеллектуальный анализ данных
Факультет управления и прикладной математики
кафедра информатики и вычислительной математики
курс: 3
квалификация: бакалавр

Семестр, формы промежуточной аттестации: 6(Весенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

практические и семинарские занятия: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 42 час.

Всего часов: 72, всего зач. ед.: 2

Количество курсовых работ, заданий: 2

Программу составил: Е.М. Лаврищева, д-р физ.-мат. наук, профессор, профессор

Программа обсуждена на заседании кафедры

15 января 2016 г.

СОГЛАСОВАНО:

Заведующий кафедрой

И.Б. Петров

Начальник учебного управления

И.Р. Гарайшина

Декан факультета

А.А. Шанин

1. Цели и задачи

Цель дисциплины

Изучение инженерных методов, процессов ЖЦ, техник проектирования и измерения процессов и продуктов. Разработчик получит знания по: применению технологии - методов и средств для управления разработкой; анализу и моделированию программных продуктов; оценке и контролю за качеством; проблемам эволюции ПО и повторному использованию программных продуктов; инструментальным средствам поддержки разработки программных систем, использованию программных продуктов и инструментальных средств поддержки разработки ПС в AppFab, в средах (IBM, VS.Net) и на ProductLineEngineeringи на фабриках программ (Greenfield, Lenz, Avdoshin, Chernetcki и др.).

Задачи дисциплины

- индустрия производства программных продуктов;
- технология программирования - методам, средствам и инструментам разработки программных компонентов, программных систем (ПС) и систем семейства (СПС);
- базовые основы современных предметно и процедурно ориентированных языков программирования (DSL, UML, Java, C#);
- методы программирования (модульного, компонентного, сервисного, агентного и др.);
- верификация, тестирование и доказательство правильности разноразовых программ, которые создаются в этих языках,
- управление коллективной разработкой, умениями выбирать процессы ЖЦ и необходимые инструментальные средства и методы, которые лучше всего подходят для данной среды разработки ПС.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Программная инженерия» включает в себя разделы, которые могут быть отнесены к вариативной части ООП.

Дисциплина «Программная инженерия» базируется на дисциплинах:

Дискретная математика;
Технология программирования;
Информатика.

Дисциплина «Программная инженерия» предшествует изучению дисциплин:

Технология открытых систем.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Освоение дисциплины направлено на формирование следующих общекультурных, общепрофессиональных и профессиональных компетенций:

В результате освоения дисциплины обучающиеся должны

знать:

- современные методы и средства технологии программирования,
- управления разработкой программных проектов,
- реинженерии и рефакторингу создаваемых программных проектов,
- новые условия, требования и возможности современных инструментальных средств, реализованных на фабриках программ на основе операционных сред типа VS.Net, IBM, Cobaiи др.

уметь:

- применять методы и средства управления разработкой проектов;
- анализа и моделирования программных продуктов;
- методов оценки и контроля качества;
- использовать современные инструментальные средства поддержки разработки программных систем и семейств систем.

владеть:

- методами и средствами управления разработкой проектов,
- методами оценки контроля качества,
- современными инструментами поддержки разработки программных систем и семейств систем.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Виды учебных занятий, включая самостоятельную работу				
		Лекции	Практические и семинарские занятия	Лаборат. работы	Задания, курсовые работы	Самост. работа
1	Дисциплины программной инженерии и области ядра знаний - SWEBOOK.	2				1
2	Стандарт и модели жизненного цикла.	2				2
3	Требования к программным системам.	2				2
4	Методы объектного анализа и моделирования	2				3
5	Прикладные и теоретические методы программирования	2				3
6	Методы доказательства, верификации и тестирования программ	2				3
7	Интерфейсы, взаимодействие, эволюция (реинженерия, рефакторинг, реверсная инженерия) программ и данных.	2				3
8	Модели качества и надежности программных систем	2				3
9	Методы управления программным проектом	2				3
10	Проблематика сборочного программирования программных систем	2				3

11	Основы построения технологических линий и средств автоматизации программных систем и семейств	2				3
12	Компоненты повторного использования. Reusability.	2				3
13	Фабрики программ, Product line SEI.	2				3
14	Индустрия программного обеспечения	2				3
15	Облачные вычисления	1				2
16	Электронное обучение программной инженерии с помощью собственных сайтов: Http://programsfactory.univ.kiev.ua , Http://sestudy.edu-ua.net	1				2
Итого часов		30				42
Подготовка к экзамену		0 час.				
Общая трудоёмкость		72 час., 2 зач.ед.				

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 6 (Весенний)

1. Дисциплины программной инженерии и области ядра знаний - SWEBOK.

Дисциплины программной инженерии и области ядра знаний - SWEBOK.

2. Стандарт и модели жизненного цикла.

Стандарт и модели жизненного цикла.

3. Требования к программным системам.

Требования к программным системам.

4. Методы объектного анализа и моделирования

Методы объектного анализа и моделирования

5. Прикладные и теоретические методы программирования

Прикладные и теоретические методы программирования

6. Методы доказательства, верификации и тестирования программ

Методы доказательства, верификации и тестирования программ

7. Интерфейсы, взаимодействие, эволюция (реинженерия, рефакторинг, реверсная инженерия) программ и данных.

Интерфейсы, взаимодействие, эволюция (реинженерия, рефакторинг, реверсная инженерия) программ и данных.

8. Модели качества и надежности программных систем

Модели качества и надежности программных систем

9. Методы управления программным проектом

Методы управления программным проектом

10. Проблематика сборочного программирования программных систем

Проблематика сборочного программирования программных систем

11. Основы построения технологических линий и средств автоматизации программных систем и семейств

Основы построения технологических линий и средств автоматизации программных систем и семейств

12. Компоненты повторного использования. Reusability.

Компоненты повторного использования. Reusability.

13. Фабрики программ, Product line SEI.

Фабрики программ, Product line SEI.

14. Индустрия программного обеспечения

Индустрия программного обеспечения

15. Облачные вычисления

Облачные вычисления

16. Электронное обучение программной инженерии с помощью собственных сайтов: [Http://programsfactory.univ.kiev.ua](http://programsfactory.univ.kiev.ua) , [Http://sestudy.edu-ua.net](http://sestudy.edu-ua.net)

Электронное обучение программной инженерии с помощью собственных сайтов: [Http://programsfactory.univ.kiev.ua](http://programsfactory.univ.kiev.ua) , [Http://sestudy.edu-ua.net](http://sestudy.edu-ua.net)

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Необходимое оборудование для лекций: компьютер, мультимедийное оборудование (проектор, звуковая система) и Интернет

Обеспечение самостоятельной работы – создание веб-сайтов силами студентов [Http://programsfactory.univ.kiev.ua](http://programsfactory.univ.kiev.ua), [Http://sestudy.edu-ua.net](http://sestudy.edu-ua.net) программирование (C#, Java, VBasic, Html, XML и др.) и отладка jПОО главных страниц сайтов и внутренних программ в указанных языках для создания и использования ТЛ производства программ в средах MicrosoftVisualStudioTeams, Corba, VIBM, и др..

6. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

Основная литература

1. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование. Киев. Наукова Думка, 1991, 209 с.
2. ИанСоммервил. Инженерия программного обеспечения. 6 -издание.–Москва–Санкт–Петербург–Киев, 2002.–623 с.
3. Лаврищева Е.М. Методы программирования. Теория, инженерия, практика-Киев, Наукова думка.- 2006.-451 с.
4. Андон Ф.И., Коваль Г.И., Коротун Т.М., Лаврищева Е.М., Суслов В.Ю. Основы инженерии качества программных систем.– К: Второе изд. Академперіодика, 2007.–680с.
5. Pfleeger S.L. Software Engineering.Theory and practice. – Printice Hall: Upper Saddenle River, New Jersey 07458, 1998. – 576р.
6. Лаврищева Е.М., Петрухин В.А. Методы и средства программного обеспечения. М.: МинобразованияРФ., 2007.-415 с.
7. Саядян, Д.Берет, Н.Мид. Мифы о программной инженерии.– М.: Открытые системы. – февраль 2003.–с.39–43 или www.osmag.ru
8. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование. Основы индустрии программных продуктов.- Киев.- Наукова думка, Киев.- 371 с.
9. Лаврищева Е.М. Software Engineering компьютерных систем. Парадигмы, Технологии, CASE-средства программирования.-Научная Думка, 2014.-284с.

Дополнительная литература

1. Лаврищева Е.М., Слабоспицкая О.А. Подход к экспертному оцениванию в программной инженерии // Кибернетика и системный анализ. – 2009. – № 4. – С. 151–168.
2. Лаврищева Е.М. Сборочное программирование. Теория и практика //Кибернетика и системный анализ. – 2009. – №6. – С. 3 – 12.
3. Лаврищева К.М., Слабоспицкая О.О., Коваль Г.И., Колесник А.О. Теоретичні аспекти керування варіабельністю в сімействах програмних систем. – Вісник КГУ, сер. фіз.–мат.наук. – 2011. – №1. –С. 151–158.
4. Lavrischeva Ekaterina, Ostrovski Alexei and Radetskyi Igor. Approach to E–Learning Fundamental Aspects of Software Engineering //ICTERI–2012 <http://senldogo0039.springer-sbm.com/ocs/home/ICTERI2012>
5. Ekaterina Lavrischeva, Andrey Stenyashin, Andrii Kolesnyk, Object-Component Development of Application and Systems. Theory and Practice, Journal of Software Engineering and Applications, 2014, 7, Published Online August 2014 in SciRes <http://www.scirp.org/journal/jsea>

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

1. <http://www.swebok.org.html>
2. <http://computer.org/education/cc2001> или <http://se.math.spbu.ru/cc2001>- русский вариант

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

1. <http://www.swebok.org.html>
2. <http://computer.org/education/cc2001> или <http://se.math.spbu.ru/cc2001>- русский вариант

9. Методические указания для обучающихся по освоению дисциплины

Студент, изучающий курс должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать основные определения понятия, аксиомы, методы доказательств.

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- выполнение курсового проекта;
- подготовка к зачету.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Показателем владения материалом служит умение решать задачи. Для формирования умения применять теоретические знания на практике студенту необходимо решать как можно больше задач. При решении задач каждое действие необходимо аргументировать, ссылаясь на известные теоретические сведения.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к лектору.

10. Фонд оценочных средств для проведения промежуточной аттестации по итогам обучения

Приложение

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ
ПО ДИСЦИПЛИНЕ**

по направлению: Прикладные математика и физика (бакалавриат)
профиль подготовки: Компьютерные технологии и интеллектуальный анализ данных
Факультет управления и прикладной математики
кафедра информатики и вычислительной математики
курс: 3
квалификация: бакалавр

Семестр, формы промежуточной аттестации: 6(Весенний) - Дифференцированный зачет

Разработчик: Е.М. Лаврищева, д-р физ.-мат. наук, профессор, профессор

1. Компетенции, формируемые в процессе изучения дисциплины

Освоение дисциплины направлено на формирование у обучающегося следующих общекультурных (ОК), общепрофессиональных (ОПК) и профессиональных (ПК) компетенций:

2. Показатели оценивания компетенций

В результате изучения дисциплины «Программная инженерия» обучающийся должен:

знать:

- современные методы и средства технологии программирования,
- управления разработкой программных проектов,
- реинженерии и рефакторингу создаваемых программных проектов,
- новые условия, требования и возможности современных инструментальных средств, реализованных на фабриках программ на основе операционных сред типа VS.Net, IBM, Cobaiи др.

уметь:

- применять методы и средства управления разработкой проектов;
- анализа и моделирования программных продуктов;
- методов оценки и контроля качества;
- использовать современные инструментальные средства поддержки разработки программных систем и семейств систем.

владеть:

- методами и средствами управления разработкой проектов,
- методами оценки контроля качества,
- современными инструментами поддержки разработки программных систем и семейств систем.

3. Перечень типовых контрольных заданий, используемых для оценки знаний, умений, навыков

1. Предмет и цель курса программной инженерии .Дисциплины программной инженерии. Сущность технологии программирования.
2. Модели, методы и средства инженерии программирования программной инженерии.
3. Модели жизненного цикла. Стандарт ISO/IEC 12207 - Процессы ЖЦ. Задачи процессов.
4. Методы программирования: структурный, ООП, UML, компонентный, сервисный и др.
5. Характеристика этих методов.
6. Методы проектирования в генерирующем программировании.
7. Инженерия применений, доменов, семейств систем. Общая их характеристика.
8. Методы и принципы представления инженерных направлений проектирования.
9. Инженерия семейств программных систем.
10. Сущность компонентов повторного использования (КПИ)
11. Языки описания специфики предметных областей – DSL..
12. Моделирование архитектур системы средствами DSL. Методы трансформации и генерации описаний программ в другие языки.
13. Подходы к преобразованию описания предметной области в DSL к языкам программирование.
14. Типы моделей для разработки предметной области -MDA, MDD, PIM, PSM и др.
15. Схемы трансформации этих моделей друг к другу.

16. Эквивалентность трансформаций разных видов моделей для многих предметных областей.
17. Аксиомы и утверждения о перестройках описаний программ в языках программирования в эквивалентные.
18. Методы эволюции ПС (рефакторинг, реинженерия).
19. Методы верификации, тестирования ПС
20. Методы доказательства программ и КПВ.
21. Теоретические методы программирования (композиционное, дескрипторное, алгоритмическое и др.)
22. Основные понятия, задачи и методы решения. Проблемы оценивания программных продуктов
23. Современные методы экспертиз и инспекций программе. Комбинированные методы проверки программ.
24. Математический аппарат принятия решений относительно разработки продукта (экспертные и байесовские методы).
25. Анализ подходов к принятию решений в процессе разработки ПП.
26. Технические, программные средства. Структура служб разработчиков ПП .
27. Перечень служб в разработке ПП соответственно стандарту ISO/IEC 12207.
28. Задачи из организации служб качества
29. Верификации, тестирования и др.
30. Изучение опыта работы оффшорных организаций, связанных с разработкой ПП. Теория измерения. Метрики и меры программного продукта.
31. Стандарты измерения. Классификация мер и метрик. Измерение программ, применений, семейств программ.
32. Теория оценивания процессов ЖЦ Методы оценки размера и сроков ПС
33. Обзор современных методов оценки ПП в статике и динамике.
34. Задача оценивания ПП по методу FRA. Оценка риска, размера, расходов, стоимости и сроков.
35. Подходы к построению ТЛ.
36. Американский вариант линий – SEI Product line
37. Методы индустрии программных продуктов.
38. Фабрики программ. Суть и структура.
39. Работа с собственной фабрикой программ и артефактов в КНУ
<http://programsfactory.univ.kiev.ua>
40. Современные основы построения инструментально-технологических комплексов производства программ и КПИ в рамках семейства ПС.

4. Критерии оценивания

Оценка «отлично (10)» выставляется обучающемуся, если показавшему всесторонние, систематизированные, глубокие знания предмета и в ходе беседы он верно и детально ответил на четыре (4) произвольных вопроса из выше приведенного перечня. Детальный ответ предполагает верные ответы на все уточняющие вопросы. Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «отлично (9)» выставляется обучающемуся, если в ходе беседы он верно, но не исчерпывающее детально ответил на четыре (4) произвольных вопроса из выше приведенного перечня (мог не ответить на некоторые уточняющие вопросы). Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «отлично (8)» выставляется обучающемуся, если в ходе беседы он верно, но не исчерпывающее детально ответил на четыре (4) произвольных вопроса из выше приведенного перечня (не ответил на уточняющие вопросы).

Оценка «хорошо (7)» выставляется обучающемуся, если в ходе беседы он верно и достаточно детально ответил на три (3) произвольных вопроса из выше приведенного перечня. Детальный ответ предполагает верные ответы на все уточняющие вопросы. Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «хорошо (6)» выставляется обучающемуся, если в ходе беседы он верно, но не исчерпывающее детально ответил на три (3) произвольных вопроса из выше приведенного перечня (не ответил на некоторые уточняющие вопросы). Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «хорошо (5)» выставляется обучающемуся, если в ходе беседы он верно и достаточно детально ответил на два (2) произвольных вопроса из выше приведенного перечня. Детальный ответ предполагает верные ответы на все уточняющие вопросы.

Оценка «удовлетворительно (4)» выставляется обучающемуся, если в ходе беседы он верно и достаточно детально ответил на один (1) произвольный вопрос из выше приведенного перечня. Детальный ответ предполагает верные ответы на все уточняющие вопросы. Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «удовлетворительно (3)» выставляется обучающемуся, если в ходе беседы он верно, но не исчерпывающее детально ответил на один (1) произвольный вопрос из выше приведенного перечня (не ответил на уточняющие вопросы).

Оценка «неудовлетворительно (2)» выставляется обучающемуся, если в ходе беседы он не смог ответить ни на один произвольный вопрос из выше приведенного перечня, но смог ответить на наводящие вопросы и вопросы с «подсказками».

Оценка «неудовлетворительно (1)» выставляется обучающемуся, если в ходе беседы он не смог ответить ни на один произвольный вопрос из выше приведенного перечня, а также ни на один наводящий вопрос.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературы.

Зачет проводится путем организации специального опроса, проводимого в устной форме.