

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

УТВЕРЖДЕНО

**Директор физтех-школы
прикладной математики и
информатики**

А.М. Райгородский

	Рабочая программа дисциплины (модуля)
по дисциплине:	Автоматизация программирования
по направлению:	Прикладная математика и информатика
профиль подготовки:	Прикладная математика и информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	1
квалификация:	магистр

Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет

Аудиторных часов: 45 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 15 час.

Самостоятельная работа: 90 час.

Всего часов: 135, всего зач. ед.: 3

Количество контрольных работ, заданий: 2

Программу составил: А.С. Хританков, канд. физ.-мат. наук, доцент

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 04.06.2020

Аннотация

Дисциплина Автоматизация программирования предусматривает изучение студентами современных методов разработки и проектирования программных систем и программно-аппаратных комплексов, основанных на применении комплексных моделей систем, генераторов программ и предметно-ориентированных языков моделирования.

1. Цели и задачи

Цель дисциплины

Дать развернутое представление о проблематике, фундаментальных концепциях, принципиальных возможностях, современном состоянии и тенденциях развития технологий разработки программного обеспечения с использованием моделей.

Задачи дисциплины

Выработать навыки работы с CASE-инструментами разработки программных систем с применением технологий MDE/MDD.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен решать актуальные задачи фундаментальной и прикладной математики	ОПК-1.1 Знает и способен использовать в профессиональной деятельности фундаментальные научные знания и новые научные принципы и методы исследований в области прикладной математики и информатики
	ОПК-1.2 Способен обобщать и критически оценивать опыт и результаты научных исследований в области профессиональной деятельности
	ОПК-1.3 Понимает междисциплинарные связи в области прикладной математики и информатики и способен их применять при решении задач профессиональной деятельности
ОПК-3 Способен разрабатывать математические модели и проводить их анализ при решении задач в области профессиональной деятельности	ОПК-3.1 Умеет анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-3.2 Владеет исследовательскими методами и способен использовать их при решении новых задач, применяя знания из различных областей науки (техники)
	ОПК-3.3 Владеет аналитическими и вычислительными методами решения, задач, понимает и учитывает на практике границы применимости получаемых решений
	ОПК-3.4 Способен самостоятельно приобретать, развивать и применять математические знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- теоретические основы разработки программного обеспечения на основе моделей;
- универсальный язык моделирования UML 2, базовый и основной UML (fUML), язык действий для UML (ALF);
- технологию исполняемого UML (Executable UML), типовые решения и принципы;
- методы преобразования моделей, методы отображения моделей в текст;
- технологию предметно-ориентированной разработки, типовые решения и принципы;
- способы применения технологий MDD в промышленной разработке ПО;
- методы тестирования, основанного на моделях.

уметь:

- ориентироваться в технологиях разработки на основе моделей, определять подходящую технологию для решения прикладных задач;
- разрабатывать предметно-ориентированные языки, трансляторы с них;
- исследовать предметную область и разрабатывать модели предметной области.

владеть:

- инструментами разработки с использованием моделей Rational Rhapsody, Eclipse DSL.

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	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Введение в технологии программной инженерии на основе моделей (MDD).	2		4	17
2	Технологии исполняемого UML.	4		4	10
3	Методы обработки моделей в MDD.	4		5	10
4	Предметно-ориентированные языки (DSL).	4		2	10
5	Применения MDSD.	4			16
6	Автоматизация разработки Automation-in-the-Small.	6			10
7	Тестирование и верификация программных систем на основе моделей (MBT).	6			17
Итого часов		30		15	90
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 2 (Весенний)

1. Введение в технологии программной инженерии на основе моделей (MDD).

Моделирование, метамоделирование. Определение MDSD, основные направления и области применения. Поколения инструментов.

2. Технологии исполняемого UML.

Основы моделирования с UML. Foundational UML (fUML), Action Language for fUML (ALF). Технология ExecutableUML, методы проектирования в OOA/RD и ExecutableUML.

3. Методы обработки моделей в MDD.

Генерация, трансляция, интерпретация. Технологии преобразования Model-to-Model и отображения в текст Model-to-Text. Алгоритмические, алгебраические и графовые модели преобразования.

4. Предметно-ориентированные языки (DSL).

Введение в Domain-Driven Design. Понятие о DSL, основные виды и паттерны разработки DSL. Методология DDD по Эвансу, эвристики и паттерны. Повсеместный язык (Ubiquitous language).

5. Применения MDSD.

Фабрики программного обеспечения Software Factories. Концепция Microsoft. Примеры Cargemini и TCS. Разработка программных систем в Motorola. Встраиваемое ПО. Линейки программных продуктов.

6. Автоматизация разработки Automation-in-the-Small.

Методы Automation-in-the-Small. Генерация UI по POJO/POCO. Автоматизация процесса сборки с помощью DSL.

7. Тестирование и верификация программных систем на основе моделей (MBT).

Введение в автоматизированное тестирование. Методы записи-воспроизведения, скрипты тестирования, табличные и DSL методы. Тестирование как сравнение с моделью. Применение и виды моделей. Критерии генерации и качества тестовых наборов.

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

Для проведения лекций необходима учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система)..

Для проведения лабораторных работ нужно:

1. Компьютерный класс с доступом в Интернет и установленными программными продуктами:
 - IBM Rational Rhapsody 8.0.5 (по академической программе IBM)
 - Eclipse Modeling Tools (EMF, Papyrus, xText, xTend) (Open Source, Eclipse Public License)
2. Предпочтительная установка в виртуальные машины, например Oracle VirtualBox.

6.Перечень рекомендуемой литературы

Основная литература

Дополнительная литература

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

1. Веб-страничка курса, <http://pps-design.org>

Дополнительная литература для самостоятельного изучения

1. Mellor S.J., Balcer M.J. Executable UML. A Foundation for Model-Driven Architecture
2. M. Fowler. Domain-Specific Languages, 2003
3. M. Utting, B. Legeard. Practical Model-Based Testing. A Tools Approach. – 2010.
- Zander, Justyna, Ina Schieferdecker, and Pieter J. Mosterman, eds. Model-based testing for embedded systems. CRC press, 2017.
4. Roques, Pascal. Systems Architecture Modeling with the Arcadia Method: A Practical Guide to Capella. – Elsevier. – 2017.
5. Эрик Эванс. Предметно-ориентированное проектирование (DDD). Структуризация сложных программных систем. – М.: Вильямс. – 2018

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

1. Eclipse DSL Tools, xText toolkit, Eclipse Modeling Tools (open-source)
2. IBM Rational Rhapsody 8 (академическая программа от IBM)

9. Методические указания для обучающихся по освоению дисциплины (модуля)

1. Ключевой частью курса является раздел о методах преобразования и отображения моделей. Освоение данного раздела лучше всего сочетается с выполнением лабораторной работы по DSL, и необходимо для ее успешной защиты.
2. Рекомендуется выполнять и сдавать лабораторные работы в срок, так как они увязаны с лекционным материалом, поясняют его на примере.
3. Для подготовки к зачету лучше всего пользоваться материалами лекций.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладная математика и информатика
профиль подготовки:	Прикладная математика и информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	<u>1</u>
квалификация:	магистр

Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет

Разработчик: А.С. Хританков, канд. физ.-мат. наук, доцент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен решать актуальные задачи фундаментальной и прикладной математики	ОПК-1.1 Знает и способен использовать в профессиональной деятельности фундаментальные научные знания и новые научные принципы и методы исследований в области прикладной математики и информатики
	ОПК-1.2 Способен обобщать и критически оценивать опыт и результаты научных исследований в области профессиональной деятельности
	ОПК-1.3 Понимает междисциплинарные связи в области прикладной математики и информатики и способен их применять при решении задач профессиональной деятельности
ОПК-3 Способен разрабатывать математические модели и проводить их анализ при решении задач в области профессиональной деятельности	ОПК-3.1 Умеет анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-3.2 Владеет исследовательскими методами и способен использовать их при решении новых задач, применяя знания из различных областей науки (техники)
	ОПК-3.3 Владеет аналитическими и вычислительными методами решения, задач, понимает и учитывает на практике границы применимости получаемых решений
	ОПК-3.4 Способен самостоятельно приобретать, развивать и применять математические знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте

2. Показатели оценивания компетенций

В результате изучения дисциплины «Автоматизация программирования» обучающийся должен:

знать:

- теоретические основы разработки программного обеспечения на основе моделей;
- универсальный язык моделирования UML 2, базовый и основной UML (fUML), язык действий для UML (ALF);
- технологию исполняемого UML (Executable UML), типовые решения и принципы;
- методы преобразования моделей, методы отображения моделей в текст;
- технологию предметно-ориентированной разработки, типовые решения и принципы;
- способы применения технологий MDD в промышленной разработке ПО;
- методы тестирования, основанного на моделях.

уметь:

- ориентироваться в технологиях разработки на основе моделей, определять подходящую технологию для решения прикладных задач;
- разрабатывать предметно-ориентированные языки, трансляторы с них;
- исследовать предметную область и разрабатывать модели предметной области.

владеть:

- инструментами разработки с использованием моделей Rational Rhapsody, Eclipse DSL.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Темы лабораторных работ:

1. Моделирование на унифицированном языке моделирования.
2. Технологии автоматизированной разработки ПО – исполняемый UML.
3. Разработка предметно-ориентированного языка и транслятора.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Определение MDSD. Расскажите об основных препятствиях в использовании MDSD. В какой части цикла инноваций для разработки приложений Gartner по вашему мнению находится MDSD?
2. Уровни моделирования в модели MOF от OMG вы знаете? Приведите пример.
3. Поколения инструментов MDSD вы знаете, опишите их, в чем различия? Приведите примеры.
4. Сравните схемы состояний UML 2 и конечные автоматы в ExecutableUML.
5. Расскажите о структуре технологии ExecutableUML, основные элементы моделей, технология генерации приложений.
6. Стандартизация технологии исполняемого UML: fUML и Alf. Текущее состояние и реализация.
7. Подходы в трансформации моделей. Программный, шаблонный (text-to-text), трансляция, преобразования на графах.
8. Методы преобразования графов (Graphrewriting). Алгебраический SPO и DPO. Понятие о TripleGraphGrammars.
9. Определение и примеры DSL. Процесс разработки. Мета модель ECore.
10. Технология разработки DSL. Генераторы и трансляция моделей. Паттерны. Инструменты.
11. Разработка DSL. Абстрактный и конкретный синтаксис. Пример на xText.
12. Определение DDD и его место в процессе разработки. Повсеместный язык (ubiquitous language). Архитектурный каркас.
13. Основные элементы и паттерны DDD.
14. Примеры реализации MDSD в индустрии. Пример Motorola.
15. Понятие о фабриках программного обеспечения. Линейка программных продуктов. Схема фабрики.
16. Основные проблемы и критические инновации в фабриках программного обеспечения.
17. Определение тестирования. Виды тестирования. Основные методы: ручной, с записью тестов, разработка скриптов, с применением DSL.
18. Тестирование на основе моделей (MBT). Определение. Метод. Виды моделей.
19. Критерии качества тестового набора.
20. Тестирование мутантами. Основные принципы. Процедурные и ОО мутации.

Примеры билетов:

Билет 1.

1. Определение MDSD. Расскажите об основных препятствиях в использовании MDSD. В какой части цикла инноваций для разработки приложений Gartner по вашему мнению находится MDSD?
2. Определение тестирования. Виды тестирования. Основные методы: ручной, с записью тестов, разработка скриптов, с применением DSL

Билет 2.

1. Подходы в трансформации моделей. Программный, шаблонный (text-to-text), трансляция, преобразования на графах.
2. Определение DDD и его место в процессе разработки. Повсеместный язык (ubiquitous language). Архитектурный каркас.

Критерии оценивания

отлично (10) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

отлично (9) - выставляется студенту, показавшему свободное оперирование знаниями учебной программы дисциплины, выполнение заданий творческого характера.

отлично (8) - выставляется студенту, показавшему владение программным учебным материалом с наличием несущественных ошибок в действиях, самостоятельно исправляемых учащимся.

хорошо (7) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускается в ответе или в решении задач некоторые неточности.

хорошо (6) - выставляется студенту если он осознает воспроизведение программного учебного материала, в том числе и различной степени сложности, с несущественными ошибками, затруднения в применении отдельных навыков.

хорошо (5) - выставляется студенту если теоретическое содержание освоено не полностью, некоторые практические навыки сформированы недостаточно, в некоторых случаях были допущены ошибки.

удовлетворительно (4) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации.

удовлетворительно (3) - выставляется студенту в случае большого количества недочетов и неправильных ответов, а также пассивной работе в ходе занятий, многие учебные задания не выполнены.

неудовлетворительно (2) - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

неудовлетворительно (1) - выставляется студенту, который не освоил теоретическое и практическое содержание курса, все выполненные учебные задания содержат грубые ошибки.

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

При проведении дифференцированного зачета обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося не должен превышать одного астрономического часа.