

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Ярославский государственный университет им. П.Г. Демидова»
Кафедра теоретической информатики

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 24 » мая 2022 г.

Рабочая программа дисциплины
«Современные парадигмы программирования»

Научная специальность
1.2.3 Теоретическая информатика, кибернетика

Форма обучения

очная

Программа рассмотрена
на заседании кафедры
от 27 апреля 2022 г.,
протокол № 9

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Современные парадигмы программирования» являются: свободное ориентирование в спектре существующих моделей и парадигм программирования, знание их достоинств и недостатков; владение мультипарадигменным языком Python, функциональным языком F# и математическими пакетами Matlab и Mathematica; умение оценить применимость и эффективность различных компьютерных технологий для конкретных прикладных задач.

2. Место дисциплины в структуре ОП аспирантуры

Дисциплина «Современные парадигмы программирования» относится к вариативной части (дисциплина по выбору) ОП аспирантуры.

Для изучения данной дисциплины необходимы «входные» знания, умения, полученные в процессе обучения по программам специалитета или бакалавриата, магистратуры. Знания и умения, приобретенные обучающимися в результате изучения дисциплины, будут использоваться при выполнении научно-исследовательской работы и подготовке кандидатской диссертации.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП аспирантуры

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Знать:

- основные типы компьютерных технологий для решения прикладных математических задач, область их применимости.

Уметь:

- адаптировать типовые алгоритмы для решения конкретных задач;
- реализовывать решения на императивных и функциональных языках;
- применять математические пакеты;
- самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, расширять и углублять свое научное мировоззрение;
- порождать новые идеи и демонстрировать навыки самостоятельной научно-исследовательской работы и работы в научном коллективе;
- разрабатывать концептуальные и теоретические модели решаемых научных проблем и задач.

Владеть:

- навыками разработки программного обеспечения и решения прикладных задач с помощью математических пакетов и языков Python и F#;
- навыками использования углубленных теоретических и практических знаний в области прикладной математики и информатики;
- навыками углубленного анализа проблем, постановки и обоснования задач научной и проектно-технологической деятельности;
- проводить семинарские и практические занятия с обучающимися, а также лекционные занятия спецкурсов по профилю специализации;
- навыками работы в международных проектах по тематике специализации;
- управлять проектами (подпроектами), планировать научно-исследовательскую деятельность, анализировать риски, управлять командой проекта.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зач. ед., 108 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	Форма промежуточной аттестации (по семестрам)
			Контактная работа						
1.	Раздел 1	2	3			1		49	
2.	Раздел 2	2	9			1		49	Самостоятельная работа №1
	Всего за 2 семестр		12			2		94	Зачет
1.	Раздел 3	3	6	3				26	Самостоятельная работа №2
2.	Раздел 4	3	6	3				26	Самостоятельная работа №3
	Всего за 3 семестр		12	6		2		52	Зачет
	Всего		24	12		4		104	

Содержание разделов дисциплины:

Раздел 1. Обзор парадигм программирования. Модели программирования: объектно-ориентированная, процессно-ориентированная, функциональная. Коллективная разработка ПО. Системы контроля версий, система SVN. Разделение интерфейса и вычислительного ядра. Тестирование. Модульные тесты. Регрессионные тесты. Экстремальное программирование, достоинства и недостатки. Непрерывная интеграция.

Раздел 2. Мультипарадигменный язык программирования Python. Применение в задачах прикладной математики и в web-программировании. Управление потоком, интроспекция. Передача аргументов. Встроенные функции. getattr(). Фильтрация списков. Логические „и” и „или”. Лямбда-функции. Итераторы и генераторы. wxPython, кросс-платформенные графические интерфейсы. Библиотеки для прикладных математических задач: NumPy, SciPy. Использование языка Python в web-приложениях. WSGI. Подключение к серверу Apache через mod_wsgi. Фреймворки web-программирования для языка Python.

Раздел 3. Функциональное программирование. Ключевые отличия от императивного программирования. Язык программирования F#. Рекурсия, неизменяемые значения. Вариантные типы данных, численные методы на F#.

Раздел 4. Функциональное и императивное программирование в пакетах прикладного математического ПО. Введение в Matlab и Mathematica. Работа с матрицами, управление потоком, линейная алгебра, интегрирование, дифференциальные уравнения.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

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

В процессе осуществления образовательного процесса используются:

– для формирования текстов материалов для промежуточной и текущей аттестации, для разработки документов, презентаций, для работы с электронными таблицами - программы OfficeStd 2013 RUS OLP NL Acdmc 021-10232, LibreOffice (свободное), издательская система LaTeX;

- компиляторы и интерпретаторы высокоуровневых языков программирования Python и F#, Matlab и Mathematica;

– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

а) основная:

1. Д.С. Глызин, Д.С. Кащенко. Параллельное и функциональное программирование. Методические указания. Яросл. гос. ун-т им. П.Г. Демидова. — Ярославль:ЯрГУ, 2009.

2. Д.В. Глазков. Пакеты прикладных математических программ. Методические указания. Яросл. гос. ун-т им. П.Г. Демидова. — Ярославль:ЯрГУ, 2009.

б) дополнительная:

1. Марк Пилгрим. Вглубь языка Питон (пер. с англ. Денис Откидач)
<http://ru.diveintopython.org>

2. Harrop, Jon D. F# for Scientists / J.D. Harrop. – John Wiley & Sons, Inc., Hoboken, New Jersey, 2008

3. Using Matlab. 1999, The Mathworks, Inc

в) ресурсы сети «Интернет»

1. Python Programming Language – Official Website: <http://python.org>

2. F# at Microsoft Research:

<http://research.microsoft.com/enus/um/cambridge/projects/fsharp/>

3. Matlab Central: <http://www.mathworks.com/matlabcentral/>

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

- специальные помещения:

-учебные аудитории для проведения занятий лекционного типа и практических занятий
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
-помещения для самостоятельной работы;
-помещения для хранения и профилактического обслуживания технических средств обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий– списочному составу группы обучающихся.

- фонд библиотеки.
- компьютерный класс с компьютерами на многоядерных процессорах с графическими картами, поддерживающими технологию CUDA и доступом к университетскому вычислительному кластеру.

Автор(ы) : профессор кафедры теоретической информатики, д.ф.-м.н., Башкин В.А.

**Приложение №1 к рабочей программе дисциплины
«Современные парадигмы программирования»
Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине**

1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

1.1. Контрольные задания и иные материалы, используемые в процессе текущей аттестации

Задания для самостоятельной работы

Самостоятельная работа № 1.

Реализовать на языке Python с применением библиотек wxPython и SciPy инструмент для численного решения дифференциальных уравнений с запаздыванием.

Самостоятельная работа № 2.

На языке F# реализовать поиск в бинарном дереве цепочки вершин с одинаковыми значениями.

Самостоятельная работа № 3.

В пакете Matlab реализовать алгоритм вычисления старшего ляпуновского показателя системы ОДУ.

Список заданий к зачету

Теоретическая часть:

1. Обзор парадигм программирования. Модели программирования: объектно-ориентированная, процессно-ориентированная, функциональная. Коллективная разработка ПО. Системы контроля версий, система SVN. Разделение интерфейса и вычислительного ядра. Тестирование. Модульные тесты. Регрессионные тесты. Экстремальное программирование, достоинства и недостатки. Непрерывная интеграция.

2. Мультипарадигменный язык программирования Python. Применение в задачах прикладной математики и в web-программировании. Управление потоком, интроспекция. Передача аргументов. Встроенные функции. getattr(). Фильтрация списков. Логические „и” и „или”. Лямбда-функции. Итераторы и генераторы. wxPython, кросс-платформенные графические интерфейсы. Библиотеки для прикладных математических задач: NumPy, SciPy. Использование языка Python в web-приложениях. WSGI. Подключение к серверу Apache через mod_wsgi. Фреймворки web-программирования для языка Python.

3. Функциональное программирование. Ключевые отличия от императивного программирования. Язык программирования F#. Рекурсия, неизменяемые значения. Вариантные типы данных, численные методы на F#.

4. Функциональное и императивное программирование в пакетах прикладного математического ПО. Введение в Matlab и Mathematica. Работа с матрицами, управление потоком, линейная алгебра, интегрирование, дифференциальные уравнения.

Практическая часть:

1. Используя лямбда-функции языка Python, реализовать программу, интерпретирующую задаваемую пользователем функцию одного переменного и выводящую ее график

2. Создать модульный тест для данной программы.

3. На языке F# реализовать бинарное дерево с целыми числами в узлах и написать функцию поиска всех вершин со значениями из заданного промежутка

Тест для самопроверки по результатам освоения дисциплины

Компетенция ПК-1

1. Назовите три ключевых особенности объектно-ориентированного программирования:

- А) наследование
- Б) либертарианство
- В) имманентность
- Г) кооптация
- Д) актуализация
- Е) полиморфизм
- Ж) трансцендентность
- З) инкапсуляция
- И) ковариантность

2. Что не является системой контроля версий:

- А) Subversion (SVN)
- Б) Git
- В) Mercurial
- Г) Wordpress

3. В языке Python запись $a = [1, 2]$ создает

- А) список
- Б) кортеж
- В) массив
- Г) строку

4. В языке Python при подключенной библиотеке NumPy код

```
a = np.array([1, 2, 3], float)
```

```
b = np.array([0, 1, 1], float)
```

```
np.dot(a, b)
```

возвращает

- А) 5.0
- Б) nan
- В) pi
- Г) $\exp(2.14)$

Компетенция ПК-2

5. Функциональное программирование описывает процесс вычисления как

- А) последовательное изменение состояний
- Б) вычисление значений функций в математическом понимании

6. Что делает следующий код

```
let fac n = List.fold (*) 1 [1..n]
```

на языке F#

- А) считает сумму чисел от 1 до n
- Б) вычисляет $n!$
- В) вычисляет скалярное произведение векторов $(1, \dots, n)$ на $(n, \dots, 1)$

7. Выберите основной элемент данных системы Matlab

- А) класс
- Б) факт
- В) массив

8. Какая команда в Mathematica используется для раскрытия скобок

- А) Cancel
- Б) Factor
- В) Expand

Вопрос №	Правильный ответ	Вопрос №	Правильный ответ
1	А, Е, З	5	Б
2	Г	6	Б
3	А	7	В
4	А	8	В

Оценка сформированности компетенций

Компетенции	Номера вопросов	Уровень формирования	Количество правильных ответов, критерии
ПК-1	1-4	Пороговый	Не менее 2
		Продвинутый	Не менее 3
		Высокий	Не менее 4
ПК-2	5-8	Пороговый	Не менее 2
		Продвинутый	Не менее 3
		Высокий	Не менее 4

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Профессиональные компетенции						
ПК-1	Самостоятельная работа №1, 2, 3. Зачет.	1-4	<p>1. Знание основных типов компьютерных технологий для решения прикладных математических задач, область их применимости.</p> <p>2. Умение адаптировать типовые алгоритмы для решения конкретных задач</p> <p>3. Умение реализовывать решения на императивных и функциональных языках.</p> <p>4. Умение применять математические пакеты.</p> <p>5. Умение</p>	<p>1. Знание основных типов компьютерных технологий для решения прикладных математических задач, область их применимости.</p> <p>2. Умение адаптировать типовые алгоритмы для решения конкретных задач</p> <p>3. Умение реализовывать решения на императивных и функциональных языках.</p> <p>4. Умение применять математические пакеты.</p>	<p>1. Знание основных типов компьютерных технологий для решения прикладных математических задач, область их применимости.</p> <p>2. Умение адаптировать типовые алгоритмы для решения конкретных задач</p> <p>3. Умение реализовывать решения на императивных и функциональных языках.</p> <p>4. Умение применять математические пакеты.</p> <p>5. Умение самостоятельно приобретать с помощью</p>	<p>1. Знание основных типов компьютерных технологий для решения прикладных математических задач, область их применимости.</p> <p>2. Умение адаптировать типовые алгоритмы для решения конкретных задач</p> <p>3. Умение реализовывать решения на императивных и функциональных языках.</p> <p>4. Умение применять математические пакеты.</p> <p>5. Умение разрабатывать концептуальные и теоретические модели решаемых научных проблем и задач.</p> <p>6. Умение самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания</p>

			<p>разрабатывать концептуальные и теоретические модели решаемых научных проблем и задач.</p> <p>6. Умение самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, расширять и углублять свое научное мировоззрение;</p> <p>7. Умение порождать новые идеи и демонстрировать навыки самостоятельной научно-исследовательской</p>		<p>информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, расширять и углублять свое научное мировоззрение;</p> <p>6. Умение порождать новые идеи и демонстрировать навыки самостоятельной научно-исследовательской работы и работы в научном коллективе.</p>	<p>и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, расширять и углублять свое научное мировоззрение;</p> <p>7. Умение порождать новые идеи и демонстрировать навыки самостоятельной научно-исследовательской работы и работы в научном коллективе.</p>
--	--	--	--	--	--	--

			работы и работы в научном коллективе.			
ПК-2	Самостоятельная работа №1, 2, 3. Зачет.	1-4	<p>1. Владение навыками разработки программного обеспечения и решения прикладных задач с помощью математических пакетов и языков Python и F#.</p> <p>2. Владение навыками использования углубленных теоретических и практических знаний в области прикладной математики и информатики.</p> <p>3. Владение навыками углубленного анализа проблем, постановки и обоснования задач научной и проектно-технологической деятельности.</p>	<p>1. Владение навыками разработки программного обеспечения и решения прикладных задач с помощью математических пакетов и языков Python и F#.</p> <p>2. Владение навыками использования углубленных теоретических и практических знаний в области прикладной математики и информатики.</p> <p>3. Владение навыками углубленного анализа проблем, постановки и обоснования задач научной и проектно-технологической</p>	<p>1. Владение навыками разработки программного обеспечения и решения прикладных задач с помощью математических пакетов и языков Python и F#.</p> <p>2. Владение навыками использования углубленных теоретических и практических знаний в области прикладной математики и информатики.</p> <p>3. Владение навыками углубленного анализа проблем, постановки и обоснования задач научной и проектно-технологической деятельности.</p> <p>4. Умение проводить семинарские и практические занятия с обучающимися, а</p>	<p>1. Владение навыками разработки программного обеспечения и решения прикладных задач с помощью математических пакетов и языков Python и F#.</p> <p>2. Владение навыками использования углубленных теоретических и практических знаний в области прикладной математики и информатики.</p> <p>3. Владение навыками углубленного анализа проблем, постановки и обоснования задач научной и проектно-технологической деятельности.</p> <p>4. Умение проводить семинарские и практические занятия с обучающимися, а также лекционные занятия спецкурсов по профилю</p>

			<p>4. Умение проводить семинарские и практические занятия с обучающимися, а также лекционные занятия спецкурсов по профилю специализации</p> <p>5. Владение навыками работы в международных проектах по тематике специализации.</p> <p>6. Умение управлять проектами (подпроектами), планировать научно-исследовательскую деятельность, анализировать риски, управлять командой проекта.</p>	<p>деятельности.</p>	<p>также лекционные занятия спецкурсов по профилю специализации</p> <p>5. Владение навыками работы в международных проектах по тематике специализации.</p>	<p>специализации</p> <p>5. Владение навыками работы в международных проектах по тематике специализации.</p> <p>6. Умение управлять проектами (подпроектами), планировать научно-исследовательскую деятельность, анализировать риски, управлять командой проекта.</p>
--	--	--	--	----------------------	--	--

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;

- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка. Для дисциплин, изучаемых в течение нескольких семестров, оценка может выставляться не только по окончании ее освоения, но и в промежуточных семестрах. Вид оценки («зачтено», «незачтено») определяется рабочей программой дисциплины в соответствии с учебным планом.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Приложение №2 к рабочей программе дисциплины «Современные парадигмы программирования»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Современные парадигмы программирования» являются лекции, причем в достаточно большом объеме. Это связано с тем, что в основе дисциплины лежит как особый математический аппарат, с помощью которого решаются довольно сложные и громоздкие задачи, так и использование современных высокоуровневых информационных технологий. По большинству тем предусмотрен большой объем самостоятельных занятий, на которых происходит закрепление лекционного материала путем применения его к конкретным задачам и отработка практических навыков программирования.

Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и практических занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач.

В конце первого и второго семестра изучения дисциплины студенты сдают зачет.

Освоить вопросы, излагаемые в процессе изучения дисциплины «Современные парадигмы программирования» самостоятельно студенту довольно сложно. Это связано со сложностью изучаемого материала и большим объемом курса. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра сдать зачет по итогам изучения дисциплины студенту довольно сложно.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Информационная система "Единое окно доступа к образовательным ресурсам" (<http://window.edu.ru/library>).

Целью создания информационной системы "Единое окно доступа к образовательным ресурсам" (ИС "Единое окно ") является обеспечение свободного доступа к интегральному каталогу образовательных интернет-ресурсов и к электронной библиотеке учебно-методических материалов для общего и профессионального образования.

Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ

(http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека «Книгообеспеченность»

(http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека «Книгообеспеченность» доступна в сети университета и через Личный кабинет.