

**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:	Концепции языков программирования
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Физтех-школа Прикладной Математики и Информатики кафедра компьютерной лингвистики
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Количество контрольных работ, заданий: 1

Программу составил: А.И. Андрианов

Программа обсуждена на заседании кафедры компьютерной лингвистики 05.06.2020

Аннотация

Целью курса является изучение концепций языков программирования: понимание языков «вглубь», изучение языков «вширь». Производится обзор понятий и приёмов, не являющихся частью языков, но влияющих на разработку ПО.

Задачами данного курса являются:

- освоение студентами базовых знаний в области программирования;
- понимание эволюции языков программирования и их сравнительных характеристик.

1. Цели и задачи

Цель дисциплины

изучить концепции языков программирования.

Задачи дисциплины

- освоение студентами базовых знаний в области программирования;
- понимание эволюции языков программирования и их сравнительных характеристик.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.3 Знает основные требования информационной безопасности

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- эволюцию и основные тенденции языков программирования;
- парадигмы программирования: функциональное и логическое, объектно-ориентированное программирование.

уметь:

- использовать языки Java, C#, C++, Prolog, Haskell, Ruby, Python; Ada, Algol, Fortran, Pas-cal, Lisp.

владеть:

- навыками освоения большого объема информации;
- навыками самостоятельной работы в Интернете;
- навыками выбора и использования языков программирования для решения соответствующих задач.

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

	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.
--	---

№	Тема (раздел) дисциплины	Лекции	Семинары	Лаборат. работы	Самост. работа
1	Введение.	2			
2	Управление потоком.	2			
3	Описание синтаксиса.	2			
4	Имена, связывание, области видимости.	2			
5	Выражения и операторы присваивания.	2			
6	Типы данных.	1			
7	Подпрограммы.	2			
8	Подпрограммы и абстракция управления потоком.	1			
9	Реализация подпрограмм.	2			
10	Распределение памяти.	2			
11	Поддержка программирования.	2			
12	Объектно-ориентированные языки программирования.	2			
13	Функциональные языки программирования.	2			
14	Логическое программирование.	2			
15	Параллелизм.	2			
16	Языки сценариев.	2			15
Итого часов		30			15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

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

Семестр: 8 (Весенний)

1. Введение.

1. Необходимость сравнительного изучения языков программирования.
2. Важнейшие критерии оценки языков программирования.
3. Архитектура компьютера и методологии разработки программ как факторы, оказывающие особое влияние на проектирование языков.
4. Основные методы реализации языков программирования: компиляция, интерпретация в чистом виде и гибридная реализация.

2. Управление потоком.

1. Операторы ветвления.
2. Операторы цикла.
3. Безусловный переход.
4. Защищенные команды.

3. Описание синтаксиса.

1. Общая задача описания синтаксиса.
2. Формальные методы описания синтаксиса.

3. Лексический разбор.
4. Синтаксический разбор.

4. Имена, связывание, области видимости.

1. Понятие связывания.
2. Время жизни объектов и управление памятью.
3. Правила обзора данных.

5. Выражения и операторы присваивания.

1. Арифметические выражения.
2. Преобразования типов.
3. Выражения отношений и булевы выражения.
4. Сокращенное вычисление.
5. Операторы присваивания.
6. Смешанное присваивание.

6. Типы данных.

1. Влияние типов данных на стиль языка и его область применения.
2. Элементарные типы данных в императивных языках.
3. Перечислимые и ограниченные типы, определяемые пользователем.
4. Массивы и записи.
5. Указатели.

7. Подпрограммы.

1. Определение подпрограммы.
2. Функции и процедуры.
3. Локальные переменные в подпрограммах.
4. Три модели передачи параметров: входной режим, выходной режим и входной-выходной режим.
5. Перегрузка операторов.

8. Подпрограммы и абстракция управления потоком.

1. Настраиваемые (шаблонные) подпрограммы.
2. Обработка исключительных ситуаций.
3. Сопрограммы.
4. События.

9. Реализация подпрограмм.

1. Общая семантика вызовов и возвратов.
2. Реализация «простых» подпрограмм.
3. Стек-динамические локальные переменные.

10. Распределение памяти.

1. Статическое выделение памяти.
2. Автоматическое выделение памяти.
3. Динамическое выделение памяти и сборка мусора.

11. Поддержка программирования.

1. Стандартная библиотека.
2. Ввод-вывод.
3. Взаимодействие с ОС.
4. Реализация GUI.

12. Объектно-ориентированные языки программирования.

1. Объектно-ориентированное программирование.
2. Инкапсуляция и наследование.
3. Создание и разрушение объектов.
4. Динамическое связывание методов.

13. Функциональные языки программирования.

1. История.
2. Понятия функционального программирования.
3. Язык Scheme.
4. Порядок вычислений.
5. Функции высшего порядка.
6. Теоретические основы.
7. Ограничения парадигмы функционального программирования.

14. Логическое программирование.

1. Понятия логического программирования.
2. Язык Prolog.
3. Теоретические основы.
4. Ограничения парадигмы логического программирования.

15. Параллелизм.

1. Мотивация.
2. Основы параллельного программирования.
3. Реализация синхронизации.
4. Механизмы поддержки параллелизма на уровне языка.
5. Передача сообщений.

16. Языки сценариев.

1. Обзор.
2. Области применения.
3. Языки сценариев для веб-программирования.
4. Инновационные особенности языков сценариев.

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

учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система).

6.Перечень рекомендуемой литературы

Основная литература

1. Язык программирования C++ (стандарт C++11) : Краткий курс [Текст] = A Tour of C++, [учеб. пособие для вузов] /Бьерн Страуструп ; пер. с англ. под ред. Н. Н. Мартынова. -М., БИНОМ, 2017
2. Язык программирования PYTHON [Текст] : учеб. пособие для вузов / Р. А. Сузи .— 2 изд., испр. — М. : Интернет-Ун-т Информ. Технологий : БИНОМ. Лаб. знаний, 2007 .— 326 с.

Дополнительная литература

1. C# в кратком изложении [Текст] : [учеб. пособие для вузов] / Дж. Бишоп, Н. Хорспул ; пер. с англ. К. Г. Финогенова .— М. : БИНОМ. Лаб. знаний, 2005 .— 472 с.
2. Основные концепции языков программирования [Текст] / Р. У. Себеста ; Ун-т штата Колорадо в Колорадо-Спрингс ; [пер. с англ. Д. А. Ключина, А. В. Назаренко ; под ред. Д. А.Ключина] .— 5-е изд. — М. : Вильямс, 2001 .— 672 с.

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

MS Visual Studio 2010, .NET Framework 4.0, MVC 3.0.

Электронные ресурсы, включая доступ к MS Development Network –
<http://msdn.microsoft.com/library/en-US>

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

MS Visual Studio 2010, .NET Framework 4.0, MVC 3.0.

Электронные ресурсы, включая доступ к MS Development Network –
<http://msdn.microsoft.com/library/en-US>

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

Методические рекомендации позволяют студенту оптимальным образом организовать процесс обучения. В рабочей программе приведено примерное распределение часов аудиторной и внеаудиторной нагрузки по различным темам данной дисциплины.

Для успешного освоения данной дисциплины студенту необходимо посещать лекции и семинарские занятия.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению: Информатика и вычислительная техника

профиль подготовки: Физтех-школа Прикладной Математики и Информатики
кафедра компьютерной лингвистики

курс: 4

квалификация: бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Дифференцированный зачет

Разработчик: А.И. Андрианов

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.3 Знает основные требования информационной безопасности

2. Показатели оценивания компетенций

В результате изучения дисциплины «Концепции языков программирования» обучающийся должен:

знать:

- эволюцию и основные тенденции языков программирования;
- парадигмы программирования: функциональное и логическое, объектно-ориентированное программирование.

уметь:

- использовать языки Java, C#, C++, Prolog, Haskell, Ruby, Python; Ada, Algol, Fortran, Pas-cal, Lisp.

владеть:

- навыками освоения большого объема информации;
- навыками самостоятельной работы в Интернете;
- навыками выбора и использования языков программирования для решения соответствующих задач.

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

1. Важнейшие критерии оценки языков программирования.
2. Основные методы реализации языков программирования: компиляция, интерпретация в чистом виде и гибридная реализация.
3. Операторы цикла.
4. Общая задача описания синтаксиса.
5. Время жизни объектов и управление памятью.
6. Выражения отношений и булевы выражения.
7. Влияние типов данных на стиль языка и его область применения.
8. Массивы и записи.
9. Локальные переменные в подпрограммах.
10. Три модели передачи параметров: входной режим, выходной режим и входной-выходной режим.
11. Обработка исключительных ситуаций.
12. Стек-динамические локальные переменные.
13. Динамическое выделение памяти и сборка мусора.
14. Инкапсуляция и наследование.
15. Ограничения парадигмы функционального программирования.
16. Ограничения парадигмы логического программирования.
17. Механизмы поддержки параллелизма на уровне языка.

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

1. Необходимость сравнительного изучения языков программирования.
2. Важнейшие критерии оценки языков программирования.
3. Архитектура компьютера и методологии разработки программ как факторы, оказывающие особое влияние на проектирование языков.
4. Основные методы реализации языков программирования: компиляция, интерпретация в чистом виде и гибридная реализация.
5. Общая задача описания синтаксиса.
6. Формальные методы описания синтаксиса.
7. Лексический разбор.
8. Синтаксический разбор.
9. Понятие связывания.
10. Время жизни объектов и управление памятью.
11. Правила обзора данных.
12. Арифметические выражения.
13. Преобразования типов.
14. Выражения отношений и булевы выражения.
15. Сокращенное вычисление.
16. Операторы присваивания.
17. Смешанное присваивание.
18. Определение подпрограммы .
19. Функции и процедуры.
20. Локальные переменные в подпрограммах.
21. Три модели передачи параметров: входной режим, выходной режим и входной-выходной режим.
22. Перегрузка операторов.
23. Настраиваемые (шаблонные) подпрограммы.
24. Обработка исключительных ситуаций.
25. Сопрограммы.
26. События.
27. Общая семантика вызовов и возвратов.
28. Реализация «простых» подпрограмм.
29. Стек-динамические локальные переменные.
30. Статическое выделение памяти.
31. Стандартная библиотека.
32. Ввод-вывод.
33. Взаимодействие с ОС.
34. Реализация GUI.
35. Стандартная библиотека
36. Объектно-ориентированное программирование.
37. Инкапсуляция и наследование.
38. Создание и разрушение объектов.
39. Динамическое связывание методов.
40. Понятия функционального программирования.
41. Язык Scheme.
42. Порядок вычислений.
43. Функции высшего порядка.
44. Ограничения парадигмы функционального программирования.
45. Понятия логического программирования.
46. Язык Prolog.
47. Ограничения парадигмы логического программирования.
48. Мотивация.
49. Основы параллельного программирования.

50. Реализация синхронизации.
51. Механизмы поддержки параллелизма на уровне языка.
52. Передача сообщений.
53. Языки сценариев для веб-программирования.
54. Инновационные особенности языков сценариев.

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

Во время проведения дифференцированного зачета по дисциплине "Концепции языков программирования" обучающиеся могут пользоваться программой дисциплины. Дифференцированный зачет проходит путем специального опроса проводимого в устной форме.