

**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**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Современные языки и платформы программирования
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Физтех-школа Прикладной Математики и Информатики кафедра когнитивных технологий
<b>курс:</b>	4
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Всего часов: 90, всего зач. ед.: 2

Количество контрольных работ, заданий: 1

Программу составил: Т.С. Чернов

Программа обсуждена на заседании кафедры когнитивных технологий 03.06.2020

## Аннотация

В рамках дисциплины слушатели изучают современных платформы, инструментальные средства и методики разработки ПО на языках высокого уровня. Целью курса является выработка у студентов понимания важности правильного выбора языковых и инструментальных средств, используемых при программной реализации алгоритмов, дать навыки использования современных инструментов и библиотек, познакомить с принятыми практиками разработки ПО промышленные качества.

### 1. Цели и задачи

#### Цель дисциплины

Изучение современных языков, платформ и методик разработки программного обеспечения.

#### Задачи дисциплины

- Выработать у студентов понимание важности постоянного внимания к эффективности алгоритмов, используемых при программировании и познакомить их с методами, которые могут использоваться для достижения эффективности, овладение студентами методами оценки эффективности, овладение типовыми алгоритмами, используемыми в задачах обработки информации, выработка умения применять типовые структуры данных и вырабатывать на их основе структуры, адекватные решаемым задачам.
- Подготовка к участию в научных семинарах, научно-технических конференциях и симпозиумах, составлению научных обзоров, рефератов и библиографии по тематике исследований.
- Подготовка к оказанию консалтинговых услуг по данной тематике.
- Подготовка к участию в международных проектах по тематике дисциплины.
- Подготовка к участию в разработке корпоративной политики и мероприятиях в области повышения социальной ответственности бизнеса перед обществом, включая разработку и реализацию решений, направленных на поддержку социально-значимых проектов.
- Совершенствование и расширение общенаучной базы. Повышение уровня общекультурного и нравственного совершенствования своей личности.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

Фундаментальные концепции и профессиональные результаты, системные методологии в профессиональной области; современное состояние и принципиальные возможности языков и систем программирования, основные методы быстрой сортировки, методы организации динамически изменяемых справочных систем, методы решения оптимизационных задач на графах.

уметь:

Использовать новые знания и применять их в профессиональной деятельности; использовать современные теории, методы, системы и средства прикладной математики и информационных технологий для решения научно-исследовательских и прикладных задач.

владеть:

Основами методологии научного познания и системного подхода при изучении различных уровней организации материи, информации, пространства и времени.

#### 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.	6	6		
2	Принципы ООП.	6	6		
3	Языки и платформы программирования.	6	6		8
4	Современные технологии разработки программного обеспечения.	6	6		10
5	Методология и средства проектирования программного обеспечения.	6	6		12
Итого часов		30	30		30
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

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

Семестр: 7 (Осенний)

###### 1. Введение.

Введение в предметную область. Исторический обзор методов и средств разработки программного обеспечения.

###### 2. Принципы ООП.

Структурное и объектно-ориентированное программирование. Основные принципы объектно-ориентированного подхода. Разработка класса. Выделение атрибутов и методов. Построение диаграмм. Языки C, C++. Определение и реализация классов средствами языка программирования C++. Конструкторы, деструкторы. Определение и использование полиморфных функций. Разработка иерархии классов. Создание производных классов. Типы отношений между классами и объектами. Определение и использование виртуальных функций. Унаследование и переопределение виртуальных функций. Принцип абстракции. Методы инкапсуляции информации. Разработка библиотек классов.

###### 3. Языки и платформы программирования.

Обзор языков программирования. C/C++, VisualBasicPascal, Java, C#. Программирование на языке Пролог. Компиляция и интерпретация программного кода. Скриптовые языки программирования. JavaScript, VBScript. Обзор Win32 API. Объектная модель Microsoft Office. VBA. Платформы программирования MicrosoftVisualStudio, C Builder. Delphi. Отладка программ. Строение прикладных frame-work. MFC. Интернет программирование. TCP/IP. Использование сокетов. Основы Java, Perl, Php. HTML. XML.

#### 4. Современные технологии разработки программного обеспечения.

Обзор технологий разработки программного обеспечения. COM архитектура. Язык определения интерфейсов IDL. MIDL. Разработка COM-компонент. ATL. Коллекции. Принципы разработки распределенных систем. DCOM. CORBA. MTS и COM+.

#### 5. Методология и средства проектирования программного обеспечения.

Принципы объектно-ориентированного анализа задач. Язык UML. CASE-средства. Rational Rose. Шаблоны проектирования. Принципы проектирования баз данных. ERwin. Моделирование бизнес процессов. BPwin. Выводы

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

Учебная аудитория, оснащенная компьютерами 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. Дизайн и эволюция языка C ++ [Текст] / Б. Страуструп ; пер. с англ. А. А. Слинкин .— М. : ДМК Пресс, 2014 .— 448 с.

#### Дополнительная литература

1. Объектно-ориентированное программирование в C++ [Текст] : [учеб. пособие для вузов] / Р. Лафоре ; [пер. с англ. А. Кузнецова и др.] .— 4-е изд. — СПб. : Питер, 2012 .— 928 с.

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

- <http://www.omg.org/>
- <http://www.vdmtools.jp/en/>
- <http://www.ttcn-3.org/>

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

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

### **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 когнитивных технологий

**курс:** 4

**квалификация:** бакалавр

Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет

**Разработчик:** Т.С. Чернов

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Современные языки и платформы программирования» обучающийся должен:

### знать:

Фундаментальные концепции и профессиональные результаты, системные методологии в профессиональной области; современное состояние и принципиальные возможности языков и систем программирования, основные методы быстрой сортировки, методы организации динамически изменяемых справочных систем, методы решения оптимизационных задач на графах.

### уметь:

Использовать новые знания и применять их в профессиональной деятельности; использовать современные теории, методы, системы и средства прикладной математики и информационных технологий для решения научно-исследовательских и прикладных задач.

### владеть:

Основами методологии научного познания и системного подхода при изучении различных уровней организации материи, информации, пространства и времени.

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

Перечень вопросов для промежуточного контроля:

1. Поколения языков программирования и вычислительных средств.
2. Алгоритмы и теоретические вычислительные устройства.
3. Типы языков программирования.
4. Парадигмы программирования.
5. Библиотеки и фреймворки.
6. Основные понятия ООП. Класс и экземпляр.
7. Экземпляры и классы. Отношения между классами.
8. Основные принципы объектной модели.
9. Объектно-ориентированный подход. ООА, ООД, ООП.
10. Режимы доступа класса. Модификация режима доступа при наследовании.
11. Множественное наследование.
12. Виртуальный вызов.
13. Таблица виртуальных функций, как вариант реализации виртуального вызова в C++.
14. Чистые виртуальные методы и абстрактные классы.
15. Система типов языка C++. Классы в C++. Статический контроль типов.
16. Перегрузка оператор в C++
17. Низкоуровневые средства C++. Работа с памятью. Адресная арифметика
18. Шаблоны в C++.
19. Свойства типов (traits).
20. Правила вывода типов.
21. Универсальные ссылки.
22. Семантика перемещения.

23. RTTI в C++.
24. Структура STL.
25. Основные контейнеры в STL.
26. STL: итераторы.
27. STL: аллокаторы.
28. STL: потоки.
29. STL: алгоритмы.
30. Потоки и локализация в C++.
31. Умные указатели.
32. Семантика перемещения.
33. Политики обработки ошибок.
34. Программирование по контракту.
35. Исключения в C++.
36. Использование препроцессора в C++.
37. Атомарные операции
38. Бинарные форматы хранения данных
39. Человекочитаемые форматы хранения данных
40. Обмен сообщениями.
41. Хранение конфигураций.
42. Бинарные интерфейсы.
43. Бинарные компоненты.
44. Уровни тестирования.
45. Библиотеки модульного тестирования.
46. Средства непрерывной интеграции.

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

Перечень контрольных вопросов:

1. Поколения языков программирования и вычислительных средств.
2. Алгоритмы и теоретические вычислительные устройства.
3. Типы языков программирования.
4. Парадигмы программирования.
5. Основные понятия ООП. Класс и объект.
6. Объекты и классы. Отношения между классами.
7. Основные принципы объектной модели.
8. Объектно-ориентированный подход. ООА, ООД, ООП.
9. Режимы доступа класса. Модификация режима доступа при наследовании.
10. Множественное наследование.
11. Виртуальные методы. Таблица виртуальных функций.
12. Чистые виртуальные методы и абстрактные классы.
13. Система типов языка C++. Классы в C++. Статический контроль типов.
14. Перегрузка операций в C++
15. Низкоуровневые средства C++. Работа с памятью. Адресная арифметика
16. Шаблоны.
17. RTTI
18. Структура STL.
19. Основные контейнеры в STL.
20. STL: итераторы, аллокаторы, потоки.
21. Потоки и локализация в C++
22. Основы языка Java. Система типов. Структура классов
23. Структура программы на C#. Проекты, компоновочные блоки, файлы
24. Система типов языка C#
25. Массивы и контейнеры языка C#

26. Перегрузка операций в C#
27. Классы, структуры и перечисления в C#
28. Делегаты и события в C#
29. Отладка и обработка исключительных ситуаций в C#
30. Программирование в среде Windows: основные черты Win32 API и MFC
31. TCP/IP и сокет.
32. Компоненты. Компонентное программирование.
33. Архитектура и назначение COM.
34. COM компоненты.
35. COM интерфейсы.
36. Интерфейс IUnknown.
37. Идентификация COM компонентов и интерфейсов. GUID. ProgID.
38. Структура реестра Windows. HKEY\_CLASSES\_ROOT.
39. Управление временем жизни COM-объектов.
40. Типы данных BSTR, VARIANT.
41. Механизм повторного использования в COM-архитектуре.
42. Поддержка версионности в COM-архитектуре.
43. Создание COM-объектов. Фабрики класса.
44. Работа с INPROC и OUTPROC серверами. Механизм удаленного вызова.
45. Конфигурирование удаленного сервера.
46. Смарт-указатели.
47. Механизм получения сообщений от ActiveX компонент.
48. IDL. MIDL.
49. Библиотеки типа.
50. IDispatch.
51. Маршalling.
52. Потоки в Win32. Проблема синхронизации.
53. Апартаменты в модели COM.
54. Суррогаты.
55. Обработка ошибок в COM.
56. COM исключения.

#### 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 " Современные языки и платформы программирования" обучающиеся могут пользоваться программой дисциплины. Дифференцированный зачет проходит путем специального опроса проводимого в устной форме.