

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики
(факультет / институт / филиал)

УТВЕРЖДЕНО
президиумом Ученого совета ННГУ
от 14.12.2021 г. протокол №4

Рабочая программа дисциплины

Языки и методы программирования
(наименование дисциплины (модуля))

Уровень высшего образования
бакалавриат
(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность
01.03.03 Механика и математическое моделирование
(указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы
Математическое моделирование и компьютерный инжиниринг
(указывается профиль / магистерская программа / специализация)

Квалификация (степень)
бакалавр
(бакалавр / магистр / специалист)

Форма обучения
Очная
(очная / очно-заочная / заочная)

Нижегород

2022

1. Место в структуре ООП

Дисциплины относится к обязательной части

№ варианта	Место дисциплины в учебном плане образовательной программы	Стандартный текст для автоматического заполнения в конструкторе РПД
1	Блок 1. Дисциплины (модули) Обязательная часть	Дисциплина Б1.О.04 «Языки и методы программирования» относится к обязательной части ООП направления подготовки 01.03.03 Механика и математическое моделирование.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	
ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	<p>ОПК-4.1. Знает базовые понятия информатики, информации, ее измерения, кодирования и представления в вычислительных системах, принципы сбора, хранения и обработки информации.</p> <p>ОПК-4.2. Умеет использовать знания, полученные в области компьютерных наук.</p> <p>ОПК-4.3. Владеть навыками использования информационных технологий для решения стандартных задач профессиональной деятельности</p>	<p>Знает базовые понятия информатики, информации, ее измерения, кодирования и представления в вычислительных системах, принципы сбора, хранения и обработки</p> <p>Уметь Самостоятельно решать задачи по составлению компьютерных программ</p> <p>Владеть навыками использования информационных технологий для решения при решении алгоритмических задач</p>	Вопросы к зачету

<p>ОПК-6. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</p>	<p>ОПК-6.1. Знает основы алгоритмизации и основы программирования, один или несколько языков программирования ОПК-6.2. Умеет разрабатывать алгоритмы и компьютерные программы для практического применения ОПК-6.3. Имеет практический опыт разработки алгоритмов и компьютерных программ для практического применения</p>	<p>Знать: основы алгоритмизации и основы программирования. Основы синтаксиса языков программирования C/C++</p> <p>Уметь: Осуществлять оценку сложности алгоритмов для решения задач. Писать программы на языках C/C++ для решения задач Имеет практический опыт находить и выбирать наиболее эффективные методы решения алгоритмических задач Владеет современными инструментальными вычислительными средствами.</p>	<p>Лабораторная работа Тест</p>
--	---	--	--

3. Структура и содержание дисциплины Языки и методы программирования

3.1.Трудоёмкость дисциплины

	Очная форма обучения
Общая трудоёмкость	11 З.Е
Часов по учебному плану	396
в том числе	
аудиторные занятия (контактная работа):	195
- занятия лекционного типа	64
- занятия семинарского типа	64
- занятия лабораторного типа	64
- текущий контроль знаний КСР	3
самостоятельная работа	165
Промежуточная аттестация – зачет, экзамен	36

3.2. Содержание дисциплины

<p>Наименование и краткое содержание разделов и тем дисциплины,</p> <p>форма промежуточной аттестации по дисциплине</p>	<p>Все го (час ы)</p>	В ТОМ ЧИСЛЕ				
		<p>контактная работа (работа во взаимодействии с преподавателем), часы</p> <p style="text-align: center;">из них</p>				<p>Самостоятел ьная работа студента часы</p>
		Занятия лекционного типа	Занятия семинарского	Лабораторные работы	КСР	Всего контакт ных часов

Технологии программирования. Базовые концепции ООП.	20	2	4	4		10	10	10
Классы С++. Стандартные члены класса.	24	2	6	6		14	10	10
Агрегация	30	4	6	6		16	14	14
Разработка простейших структур данных	35	4	8	8		20	15	15
Перегрузка операций.	34	4	8	8		20	14	14
Промежуточная аттестация. Зачет.	1		1					
Связанные списки	32	4	4	4		12	20	20
Бинарные деревья	32	4	4	4		12	20	20
Наследование и полиморфизм	44	8	8	8		24	20	20
Шаблоны функций и классов	36	8	8	8		24	12	12
Стандартная библиотека шаблонов	34	8	8	8		24	10	10
В т.ч. текущий контроль	2		2					
Промежуточная аттестации – Экзамен (36 часов)								

Используются активные и интерактивные образовательные технологии в форме лекций, практических занятий, лабораторных работ.

Лекция-информация. Ориентирована на изложение и объяснение студентам научной информации, подлежащей осмыслению и запоминанию.

Лекция-беседа, или «диалог с аудиторией», является наиболее распространенной и сравнительно простой формой активного вовлечения студентов в учебный процесс. Эта лекция предполагает непосредственный контакт преподавателя с аудиторией. Преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание слушателей к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей обучаемых.

К участию в лекции-беседе можно привлечь различными приемами, так, например, озадачивание слушателей вопросами в начале лекции и по ее ходу. Вопросы могут, быть информационного и проблемного характера, для выяснения мнений и уровня осведомленности по рассматриваемой теме, степени их готовности к восприятию последующего материала. Вопросы адресуются всей аудитории. Слушатели отвечают с мест. Если преподаватель замечает, что кто-то из обучаемых не участвует в ходе беседы, то вопрос можно адресовать лично тому слушателю, или спросить его мнение по обсуждаемой проблеме. Для экономии времени вопросы рекомендуется формулировать так, чтобы на них можно было давать однозначные ответы. С учетом разногласий или единодушия в ответах преподаватель строит свои дальнейшие рассуждения, имея при этом возможность, наиболее доказательно изложить очередное понятие лекционного материала.

Вопросы могут быть как простыми для того, чтобы сосредоточить внимание слушателей на отдельных аспектах темы, так и проблемные. Обучаемый, продумывая ответ на заданный вопрос, получает возможность самостоятельно прийти к тем выводам и обобщениям, которые преподаватель должен был сообщить им в качестве новых знаний,

либо понять важность обсуждаемой темы, что повышает интерес, и степень восприятия материала слушателями.

Во время проведения лекции-беседы задаваемые вопросы не должны оставаться без ответов, иначе они будут носить риторический характер, не обеспечивая достаточной активизации мышления обучаемых. Наиболее проблемные вопросы могут быть вынесены на самостоятельную работу студентов, проверку которой преподаватель осуществляет в рамках текущего контроля успеваемости и/или промежуточной аттестации.

Эффективность лекции-беседы в условиях группового обучения снижается из-за того, что не всегда удается каждого обучаемого вовлечь в двусторонний обмен мнениями.

Лекция с заранее запланированными ошибками - рассчитана на стимулирование студентов к постоянному контролю предлагаемой информации (поиск ошибки: содержательной, методологической, методической, орфографической). Предварительно студенты могут быть разбиты на малые группы. В конце лекции или на следующем занятии проводится диагностика групп и разбор сделанных ошибок.

Данный вид лекции проводится в завершение темы или раздела учебной дисциплины, когда у слушателей сформированы основные понятия и представления. Заключительный анализ ошибок развивает у слушателей теоретическое мышление.

Лекция-консультация – по типу «**вопросы—ответы—дискуссия**», является тройным сочетанием: изложение новой учебной информации лектором, постановка вопросов и организация дискуссии в поиске ответов на поставленные вопросы». Проводится в период работы обучающихся над проектной работой.

Лекция-консультация по типу «**вопросы—ответы**». Лектор отвечает в течение лекционного времени на вопросы студентов по всем разделу или всему курсу. Проводится перед защитой проектных работ и промежуточной аттестацией.

Текущий контроль успеваемости реализуется в рамках занятий семинарского и лабораторного типа. Для выполнения лабораторных работ предусматривается работа в терминал-классах.

Итоговая аттестация проходит в традиционной форме (зачет, экзамен), включающий изложение теоретических вопросов и выполнение типовых заданий.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

А. Виды самостоятельной работы студентов

Выполнение лабораторных работ на следующие темы:

Лабораторная работа "Класс Date"

Лабораторная работа «Класс Vector»

Лабораторная работа «Класс Complex»

Лабораторная работа «Иерархия геометрических фигур»

Лабораторная работа «Реализация стека на массиве»

Лабораторная работа "Вычисление арифметического выражения"

В. Образовательные материалы для студентов

- Список тем для самостоятельной проработки

1. Модульное программирование

Программы из нескольких файлов. Проекты. Заголовочные файлы. Страж включения.

Организация связи между модулями.

2. Особенности функций C++

Перегрузки функций. Аргументы функций по умолчанию.

3. Ссылки. Работа с памятью

Понятие ссылки. Использование ссылок. Динамическое выделение памяти оператором new. Освобождение памяти оператором delete.

4. Структуры в C++

Состав структур. Поля и методы. Определение функций-членов структуры

5. Понятие класса

Скрытие данных. Открытые и закрытые члены.

6. Конструкторы класса

Понятие конструктора. Назначение конструктора. Объявление и определение конструкторов. Использование конструкторов.

7. Статические члены. Ссылка на себя

Понятие статического члена класса. Объявление и определение статических членов. Статические функции-члены класса. Указатель this на объект класса. Использование указателя this.

8. Копирование объектов класса

Встроенный механизм копирования.

9. Друзья класса

Объявление и определений функций-друзей класса. Классы-друзья..

10. Управление доступом

Структуры и классы. Правила доступа.

11. Деструкторы

Синтаксис. Назначение. Использование.

12. Программы из нескольких файлов

Проекты. Заголовочные файлы. страж включения. Модули. Область действия имен. Глобальные и локальные имена. Статические имена.

13. Пространство имен

Необходимость пространства имен. Объявление пространства имен. Объявление using. Директива using, стандартные пространства имен.

14. Перегрузка операторов

Синтаксис перегрузки операторов. Правила перегрузки унарных и бинарных операторов. Использование перегруженных операторов.

15. Конструктор копирования и оператор присваивания

Недостатки встроенного механизма копирования. Проблемы с указателями - членами классов. Конструктор копирования. Случаи использования конструктора копирования.

16. Ввод и вывод

Стандартная библиотека потоковых классов ввода и вывода. Текстовые и бинарные файлы, файловый ввод и вывод.

17. Ввод и вывод пользовательских типов данных

Перегрузка операторов ввода и вывода для пользовательских типов данных.

18. Объекты как члены класса

Синтаксис вызова конструктора для объектов-членов класса. Конструкторы встроенных типов данных.

19. Наследование

Понятие наследования. Управление доступом при наследовании. Наследование и конструкторы.

20. *Полиморфизм*

Виртуальные функции. Перегрузка виртуальных функций. Вызов виртуальных функций. Раннее и позднее связывание.

21. *Абстрактные классы*

Чистая виртуальная функция. Назначение абстрактных классов. Использование абстрактных классов.

22. *Множественное наследование*

Механизм выбора функций из базовых классов. Приведение типов при множественном наследовании.

23. *Шаблоны*

Шаблоны функций. Шаблоны классов.

24. *Стандартная библиотека шаблонов*

Контейнеры. Алгоритмы. Итераторы.

25. *Обработка исключений*

Понятие исключения. Синтаксис обработки исключений.

Список вопросов к зачету

1. Ссылки. Ссылка как аргумент функции. Возврат ссылки из функции.
2. Операторы new и delete. Привести примеры использования.
3. Функции как члены структуры.
4. Встроенные функции.
5. Классы. Инкапсуляция. Члены класса. Скрытие данных. Объекты класса.
6. Конструкторы.
7. Размер класса и размер объектов класса.
8. Друзья класса. Привести пример.
9. Копирование объектов класса.
10. Структуры и классы. Правила доступа к членам класса.
11. Статические члены класса. Привести пример.
12. Ссылка на себя. Программа модификации дат.
13. Деструкторы. Деструктор в классе дат.
14. Область действия имен. Глобальные и локальные имена. Статические имена.
15. Заголовочные файлы. Страж включения.
16. Понятие стека. Реализация стека в виде массива.
17. Программа анализа правильности расстановки скобок в тексте.
18. Пространства имен.
19. Стандартное пространство имен std.
20. Перегрузка операторов.
21. Правила перегрузки операторов.
22. Программа моделирования комплексных чисел.
23. Проблемы при копировании объектов, содержащих указатели.
24. Конструктор копирования.
25. Вывод в языке C++.
26. Ввод в языке C++.
27. Перегрузка операторов ввода для пользовательских типов.
28. Перегрузка операторов вывода для пользовательских типов.
29. Файловый ввод и вывод.

Список вопросов к экзамену

1. Ссылки. Ссылка как аргумент функции. Возврат ссылки из функции.
2. Операторы new и delete. Привести примеры использования.
3. Функции как члены структуры.
4. Встроенные функции.
5. Классы. Инкапсуляция. Члены класса. Скрытие данных. Объекты класса.
6. Конструкторы.
7. Размер класса и размер объектов класса.
8. Друзья класса. Привести пример.
9. Копирование объектов класса.
10. Структуры и классы. Правила доступа к членам класса.
11. Статические члены класса. Привести пример.
12. Ссылка на себя.
13. Деструкторы. Деструктор в классе дат.
14. Область действия имен. Глобальные и локальные имена. Статические имена.
15. Заголовочные файлы. Страж включения.
16. Понятие стека. Реализация стека в виде массива.
17. Программа анализа правильности расстановки скобок в тексте.
18. Пространства имен.
19. Стандартное пространство имен std.
20. Перегрузка операторов.
21. Правила перегрузки операторов.
22. Программа моделирования комплексных чисел.
23. Проблемы при копировании объектов, содержащих указатели.
24. Конструктор копирования.
25. Вывод в языке C++.
26. Ввод в языке C++.
27. Перегрузка операторов ввода для пользовательских типов.
28. Перегрузка операторов вывода для пользовательских типов.
29. Файловый ввод и вывод.
30. Сравнение текстового и бинарного файлов.
31. Объекты как члены класса.
32. Конструкторы встроенных типов.
33. Наследование классов.
34. Управление доступом при наследовании.
35. Наследование и конструкторы
36. Производный класс личных данных.
37. Виртуальные функции.
38. Абстрактные классы
39. Вызов виртуальных функций.
40. Локальные классы.
41. Совместимость типов при наследовании.
42. Множественное наследование.
43. Класс алгебраических векторов Vector.
44. Класс прямоугольных матриц.
45. Класс систем линейных уравнений.
46. Шаблоны функций.
47. Шаблоны классов.
48. Шаблон классов векторов.
49. Шаблон классов динамических массивов.
50. Обработка исключений.

51. Стандартная библиотека шаблонов. Привести пример использования шаблонов стандартной библиотеки.
52. Динамическое создание объектов класса. Виртуальные деструкторы.

5. Фонд оценочных средств для промежуточной аттестации по дисциплине

Фонд оценочных средств включает: контрольные материалы для проведения текущего контроля в форме *задач (практических заданий и лабораторных работ), контрольных работ* и контрольные материалы для проведения промежуточной аттестации в форме вопросов и заданий к *зачётному экзамену*.

5.1. Описание шкал оценивания результатов обучения по дисциплине

Шкала оценивания сформированности компетенций		Уровень сформированности компетенций (индикатора достижения компетенций)		
		Знания	Умения	Навыки
плохо	не зачтено	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	Отсутствие владения материалом. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа
		Уровень знаний ниже минимальных требований. Имели место грубые ошибки.	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки.	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки.
удовлетворительно	зачтено	Минимально допустимый уровень знаний. Допущено много негрубых ошибок.	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами
хорошо		Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами.
очень хорошо		Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов.
отлично		Уровень знаний в объеме, соответствующем программе подготовки, без ошибок.	Продемонстрированы все основные умения, решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме.	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов.
превосходно		Уровень знаний в объеме, превышающем	Продемонстрированы все основные умения.	Продемонстрирован творческий подход к

Шкала оценивания сформированности компетенций	Уровень сформированности компетенций (индикатора достижения компетенций)		
	Знания	Умения	Навыки
	программу подготовки.	Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов	решению нестандартных задач

Шкала оценки при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно»
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения

Для оценивания результатов обучения в виде знаний используются следующие процедуры и технологии:

- письменные ответы на вопросы.

Для оценивания результатов обучения в виде умений и владений используются следующие процедуры и технологии:

- лабораторные работы, включающих постановку одной сложной учебной задачи в виде краткой формулировки действий, которые следует выполнить, и описания результата, который нужно получить.

5.2.1. Вопросы для текущего контроля

Вопрос 1.1. Технологии программирования: структурное, модульное, ООП. Общее описание.

Вопрос 1.2. Концепции ООП: абстрагирование, инкапсуляция. Поддержка в языке программирования C++. Примеры.

Вопрос 1.3. Концепции ООП: наследование. Поддержка в языке программирования C++. Примеры.

Вопрос 1.4. Концепции ООП: полиморфизм. Поддержка в языке программирования C++. Примеры.

Вопрос 1.5. Создание объектов. Конструкторы. Виды конструкторов, объявление. Реализация конструкторов по умолчанию и инициализации на примере класса `Complex`. Примеры использования.

Вопрос 1.6. Создание объектов. Конструкторы. Виды конструкторов, объявление. Реализация конструкторов копирования и преобразования типа на примере класса `Complex`. Примеры использования.

Вопрос 1.7. Удаление объектов. Деструктор. Пример реализации для класса `String`. Реализация метода `Add` для класса `Complex`.

Вопрос 1.8. Перегрузка функций в C++. Перегрузка унарных и бинарных операций. Перегрузка операций $++$ и $++$ для класса `Complex`. Примеры использования.

Вопрос 1.9. Перегрузка функций в C++. Перегрузка унарных и бинарных операций. Перегрузка операций $+$ и $+=$ для класса `Complex`. Примеры использования.

Вопрос 1.10. Перегрузка функций в C++. Перегрузка унарных и бинарных операций. Перегрузка оператора присваивания на примере класса `String`. Отличия оператора присваивания от конструктора копирования. Примеры использования.

Вопрос 1.11. Разработка класса `Vector`. Объявление (конструкторы, деструктор, перегрузка операций $+$, $-$, $+=$, $-$, $++$, $--$, $[], =$). Реализация конструкторов, деструктора, оператора присваивания.

Вопрос 1.12. Разработка класса `Vector`. Объявление (конструкторы, деструктор, перегрузка операций $+$, $-$, $+=$, $-$, $++$, $--$, $[], =$). Реализация операций $+$, $+=$, $++$, $[], =$.

Вопрос 1.13. Спецификаторы доступа в объявлении класса. Примеры. Дружественные функции на примере операции $+$ для класса `Complex`.

Вопрос 1.14. Полиморфизм на примере иерархии классов геометрических фигур. Таблица виртуальных функций. Преимущества использования полиморфизма.

Вопрос 1.15. Полиморфизм на примере иерархии классов геометрических фигур. Абстрактные классы. Виртуальный деструктор.

Вопрос 1.16. Макросы и шаблоны функций. Отличия, плюсы и минусы. Примеры.

Вопрос 1.17. Шаблоны классов. Описание. Отличия реализации шаблонного класса от обычного. Пример: динамический массив.

5.2.2. Примеры тестов для оценивания результатов формирования знаний, умений ОПК-4, ОПК-6.

1. Что из перечисленного является объектом?

Число	Цвет	Номер кредитной карты	Счет в банке
-------	------	-----------------------	--------------

Ответ: 4

2. Отношение целого и его части, приводящие к соотв. иерархии объектов

Абстракция	Агрегация	Консолидация	Типизация
------------	-----------	--------------	-----------

Ответ: 2

3. Для чего используются классы?

Для создания эффективных программ	Для разработки прототипов	Для создания объектов	Для описания шаблонов
-----------------------------------	---------------------------	-----------------------	-----------------------

Ответ: 2,3

4. Чем структура в C++ отличается от класса?

Ничем	В структуре по умолчанию все открыто	В структуре не могут использоваться функции	В структуре по умолчанию все закрыто
-------	--------------------------------------	---	--------------------------------------

Ответ: 1

Критерий оценивания результатов тестирования

Баллы, %	Оценка
99-100	Превосходно
91-98	Отлично

86-90	Очень хорошо
71-85	Хорошо
51-70	Удовлетворительно
31-50	Неудовлетворительно
0-30	Плохо

Образец экзаменационного билета на оценивание формирования компетенций ОК-7, ОПК-1, ОПК-4

Национальный исследовательский Нижегородский государственный университет

им. Н.И. Лобачевского

Институт/факультет _____ ИИТММ _____

Кафедра _____ Программная инженерия

Дисциплина _____ Языки и методы программирования _____

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Ссылки. Ссылка как аргумент функции. Возврат ссылки из функции.
2. Ввод в языке C++.

Зав. кафедрой _____

Экзаменатор _____

6. Учебно-методическое и информационное обеспечение дисциплины
а) основная литература:

1. Окулов, С.М. Программирование в алгоритмах. [Электронный ресурс] — Электрон. дан. — М. : Издательство "Лаборатория знаний", 2017. — 386 с. — Режим доступа: <http://e.lanbook.com/book/94140> — Загл. с экрана.

2. Логинова, Ф.С. Объектно-ориентированные методы программирования. [Электронный ресурс] : учеб. пособие — Электрон. дан. — СПб. : ИЭО СПбУТУиЭ, 2012. — 208 с. — Режим доступа: <http://e.lanbook.com/book/64040> — Загл. с экрана.
 3. Конова, Е.А. Алгоритмы и программы. Язык С++. [Электронный ресурс] : учеб. пособие / Е.А. Конова, Г.А. Поллак. — Электрон. дан. — СПб. : Лань, 2017. — 384 с. — Режим доступа: <http://e.lanbook.com/book/90158> — Загл. с экрана.
 4. Васильев, А.Н. Самоучитель С++ с примерами и задачами. Книга + виртуальный CD. [Электронный ресурс] : самоучитель — Электрон. дан. — СПб. : Наука и Техника, 2016. — 480 с. — Режим доступа: <http://e.lanbook.com/book/74667> — Загл. с экрана.
 5. Васильев, А.Н. Объектно-ориентированное программирование на С++. [Электронный ресурс] — Электрон. дан. — СПб. : Наука и Техника, 2016. — 544 с. — Режим доступа: <http://e.lanbook.com/book/90227> — Загл. с экрана.
 6. Бабушкина, И.А. Практикум по объектно-ориентированному программированию. [Электронный ресурс] : учеб. пособие / И.А. Бабушкина, С.М. Окулов. — Электрон. дан. — М. : Издательство "Лаборатория знаний", 2015. — 369 с. — Режим доступа: <http://e.lanbook.com/book/66121> — Загл. с экрана.
- б) дополнительная литература:
1. Комлев, Н.Ю. Объектно-Ориентированное Программирование. Хорошая книга для хороших людей. [Электронный ресурс] — Электрон. дан. — М. : СОЛОН-Пресс, 2014. — 298 с. — Режим доступа: <http://e.lanbook.com/book/64985> — Загл. с экрана.
 2. Дудецкий, В.Н. Объектно-ориентированные языки программирования: учеб. пособие: в 3 ч. Ч. I. [Электронный ресурс] : учеб. пособие — Электрон. дан. — М. : ФЛИНТА, 2015. — 48 с. — Режим доступа: <http://e.lanbook.com/book/74715> — Загл. с экрана.
 3. Дудецкий, В.Н. Объектно-ориентированные языки программирования: учеб. пособие: в 3 ч. Ч. II. [Электронный ресурс] : учеб. пособие — Электрон. дан. — М. : ФЛИНТА, 2015. — 40 с. — Режим доступа: <http://e.lanbook.com/book/74780> — Загл. с экрана.
 4. Гаврилова, И.В. Разработка приложений. [Электронный ресурс] : учеб. пособие — Электрон. дан. — М. : ФЛИНТА, 2012. — 241 с. — Режим доступа: <http://e.lanbook.com/book/44746> — Загл. с экрана.
 5. Бородихин, В.Н. Языки программирования (СИ/СИ++): учебно-методическое пособие. [Электронный ресурс] : учеб.-метод. пособие — Электрон. дан. — Омск : ОмГУ, 2013. — 200 с. — Режим доступа: <http://e.lanbook.com/book/75386> — Загл. с экрана.

в) программное обеспечение и Интернет-ресурсы

- Microsoft Visual Studio 2010-2015

7. Материально-техническое обеспечение дисциплины

Учебные аудитории, оборудованные мультимедийной техникой (компьютер, проектор, экран), для проведения занятий лекционного и семинарского типа. Для проведения лабораторных занятий необходим компьютерный класс, оснащенный персональными компьютерами по числу студентов в группе, проектором и преподавательским компьютером.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению 010303 «Механика и математическое моделирование».

Автор (ы) _____ Штанюк А.А. _____

Рецензент (ы) _____

Заведующий кафедрой _____

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики

от 01.12.2021 года, протокол № 2.