

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

УТВЕРЖДЕНО  
президиумом Ученого совета ННГУ  
протокол от  
«14» декабря 2021 г. № 4

**Рабочая программа дисциплины**

*Языки и методы программирования*

Уровень высшего образования  
*бакалавриат*

Направление подготовки / специальность

*02.03.01 Математика и компьютерные науки*

Направленность образовательной программы  
*Общий профиль*

Форма обучения  
*Очная*

Нижегород

2022 год

## 1. Место дисциплины в структуре ОПОП

Дисциплина относится к обязательной части *Б1.О.18 Языки и методы программирования*

| № варианта | Место дисциплины в учебном плане образовательной программы | Стандартный текст для автоматического заполнения в конструкторе РПД   |
|------------|--|---|
| 1          | Блок 1. Дисциплины (модули)<br>Обязательная часть          | Дисциплина Б1.О.18, «Языки и методы программирования», относится к обязательной части ООП направления подготовки 02.03.01 Математика и компьютерные науки |

## 2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

| 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**  |                                  |
| <b>ОПК-6.</b> Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения | ОПК-6.1. Знает основы алгоритмизации и основы программирования, один или несколько языков программирования  | Знать основные принципы использования программных инструментов для разработки приложений, принципы использования языка программирования высокого уровня при разработке программных систем для решения задач профессиональной деятельности. | <i>Тестирование</i>              |
|   | ОПК-6.2. Умеет разрабатывать алгоритмы и компьютерные программы для практического применения                | Уметь составлять алгоритмы для решения профессиональных задач, применять на практике подходы, алгоритмы, используемые при написании программ.  | <i>Собеседование, Задача</i>     |
|   | ОПК-6.3. Имеет практический опыт разработки алгоритмов и компьютерных программ для практического применения | Владеть инструментами для разработки программных систем, основными подходами к разработке программ для решения задач профессиональной деятельности.  | <i>Собеседование</i>             |

### 3. Структура и содержание дисциплины

#### 3.1. Трудоемкость дисциплины

|  |                             |
|--|-----------------------------|
|  | <b>очная форма обучения</b> |
| <b>Общая трудоемкость</b>                        | <b>10 ЗЕТ</b>               |
| <b>Часов по учебному плану</b>                   | <b>360</b>                  |
| <b>в том числе</b>                               |                             |
| <b>аудиторные занятия (контактная работа)</b>    | <b>195</b>                  |
| <b>самостоятельная работа</b>                    | <b>129</b>                  |
| <b>Контроль</b>                                  | <b>36</b>                   |
| <b>Промежуточная аттестация – зачет, экзамен</b> |                             |

#### 3.2. Содержание дисциплины

| Наименование и краткое содержание разделов и тем дисциплины | Всего (часы) | В том числе   |                           |                            |       |   |
|---|--------------|---|---------------------------|----------------------------|-------|---|
|   |              | Контактная работа (работа во взаимодействии с преподавателем), часы. Из них |                           |                            |       | Самостоятельная работа обучающегося, часы |
|   |              | Занятия лекционного типа  | Занятия семинарского типа | Занятия лабораторного типа | Всего |   |
| Технологии программирования. Базовые концепции ООП.         | 13           | 6   | 0                         | 0                          | 6     | 7   |
| Классы C++. Стандартные члены класса.                       | 32           | 6   | 8                         | 8                          | 22    | 10  |
| Агрегация   | 32           | 6   | 8                         | 8                          | 22    | 10  |
| Разработка простейших структур данных                       | 34           | 8   | 8                         | 8                          | 24    | 10  |
| Перегрузка операций.  | 32           | 6   | 8                         | 8                          | 22    | 10  |
| Текущий контроль (КСР)                                      | 1            |   |                           |                            |       |   |
| <b>Промежуточная аттестация – зачет</b>                     |              |   |                           |                            |       |   |
|   |              |   |                           |                            |       |   |
| Связанные списки  | 19           | 6   | 6                         | 6                          | 18    | 1   |
| Бинарные деревья  | 38           | 6   | 6                         | 6                          | 18    | 20  |
| Наследование и полиморфизм                                  | 38           | 6   | 6                         | 6                          | 18    | 20  |
| Шаблоны функций и классов                                   | 38           | 6   | 6                         | 6                          | 18    | 20  |
| Стандартная библиотека шаблонов                             | 44           | 8   | 8                         | 8                          | 24    | 20  |
| Текущий контроль (КСР)                                      | 3            |   |                           |                            |       |   |
| <b>Итоговая аттестация – экзамен</b>                        | 36           |   |                           |                            |       |   |
| <b>Итого</b>  | 360          | 64  | 64                        | 64                         | 188   | 128                                       |
|   |              |   |                           |                            |       |   |

Текущий контроль успеваемости реализуется в формах опросов на занятиях семинарского типа, лабораторного типа.

Промежуточная аттестация проходит в традиционных форма (зачет, экзамен).

#### **4. Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Языки и методы программирования»**

А. Виды самостоятельной работы студентов

- проработка тем для самостоятельной подготовки
- подготовка к промежуточной и итоговой аттестациям

В. Образовательные материалы для самостоятельной работы студентов

- Материалы курса лекций «[Основы](#) программирования на языке С»: НОУ ИНТУИТ <http://www.intuit.ru/studies/courses/43/43/info>, режим доступа – свободный

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

- 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.

**5. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю), включающий:**

**5.1. Описание шкал оценивания результатов обучения по дисциплине «Языки и методы программирования»**

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

### 5.2.1 Контрольные вопросы к зачету (2 семестр)

| Вопросы  | Код формируемой компетенции |
|--|-----------------------------|
| <ol style="list-style-type: none"> <li>1. Ссылки. Ссылка как аргумент функции. Возврат ссылки из функции.</li> <li>2. Операторы new и delete. Привести примеры использования.</li> <li>3. Функции как члены структуры.</li> <li>4. Встроенные функции.</li> <li>5. Классы. Инкапсуляция. Члены класса. Скрытие данных. Объекты класса.</li> <li>6. Конструкторы.</li> <li>7. Размер класса и размер объектов класса.</li> <li>8. Друзья класса. Привести пример.</li> <li>9. Копирование объектов класса.</li> <li>10. Структуры и классы. Правила доступа к членам класса.</li> <li>11. Статические члены класса. Привести пример.</li> <li>12. Ссылка на себя. Программа модификации дат.</li> <li>13. Деструкторы. Деструктор в классе дат.</li> <li>14. Область действия имен. Глобальные и локальные имена. Статические имена.</li> <li>15. Заголовочные файлы. Страж включения.</li> <li>16. Понятие стека. Реализация стека в виде массива.</li> <li>17. Программа анализа правильности расстановки скобок в тексте.</li> <li>18. Пространства имен.</li> <li>19. Стандартное пространство имен std.</li> <li>20. Перегрузка операторов.</li> <li>21. Правила перегрузки операторов.</li> <li>22. Программа моделирования комплексных чисел.</li> </ol> | ОПК-6                       |

|  |  |
|--|--|
| 23. Проблемы при копировании объектов, содержащих указатели.<br>24. Конструктор копирования.<br>25. Вывод в языке C++.<br>26. Ввод в языке C++.<br>27. Перегрузка операторов ввода для пользовательских типов.<br>28. Перегрузка операторов вывода для пользовательских типов.<br>29. Файловый ввод и вывод. |  |
|--|--|

### Контрольные вопросы к экзамену (3 семестр)

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



|   |  |
|---|--|
| 29. Файловый ввод и вывод.<br>30. Сравнение текстового и бинарного файлов.<br>31. Объекты как члены класса.<br>32. Конструкторы встроенных типов.<br>33. Наследование классов.<br>34. <a href="#">Управление доступом при наследовании</a> .<br>35. <b>Ошибка! Закладка не определена.</b><br>36. Производный класс личных данных.<br>37. Виртуальные функции.<br>38. Абстрактные классы<br>39. Вызов виртуальных функций.<br>40. Локальные классы.<br>41. Совместимость типов при наследовании.<br>42. Множественное наследование.<br>43. Класс алгебраических векторов Vector.<br>44. Класс прямоугольных матриц.<br>45. Класс систем линейных уравнений.<br>46. Шаблоны функций.<br>47. Шаблоны классов.<br>48. Шаблон классов векторов.<br>49. Шаблон классов динамических массивов.<br>50. Обработка исключений.<br>51. Стандартная библиотека шаблонов. Привести пример использования шаблонов стандартной библиотеки.<br>52. Динамическое создание объектов класса. Виртуальные деструкторы. |  |
|---|--|

### 5.2.2. Типовые тестовые задания для оценки сформированности компетенции ОПК-6

1. Что из перечисленного является объектом?

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

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

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

3. Для чего используются классы?

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

**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 |
|-------|--------------------------------------|---|--------------------------------------|

**5. Конструктор – это функция, которая**

|                                  |                                  |                                 |                                      |
|----------------------------------|----------------------------------|---------------------------------|--------------------------------------|
| Вызывается при запуске программы | Вызывается при компиляции класса | Вызывается при создании объекта | Вызывается при использовании объекта |
|----------------------------------|----------------------------------|---------------------------------|--------------------------------------|

**6. Сколько в классе может быть деструкторов?**

|      |     |     |                |
|------|-----|-----|----------------|
| Один | Два | Три | Сколько угодно |
|------|-----|-----|----------------|

**7. Что может быть полем класса?**

|        |                  |           |           |
|--------|------------------|-----------|-----------|
| Массив | Экземпляр класса | Константа | Выражение |
|--------|------------------|-----------|-----------|

**8. Может ли метод быть описан в объявлении класса?**

|            |              |               |  |
|------------|--------------|---------------|--|
| Да, каждый | Нет, никакой | Да, некоторые |  |
|------------|--------------|---------------|--|

**9. Вызов методов класса возможен**

|                        |                  |                           |  |
|------------------------|------------------|---------------------------|--|
| Через экземпляр класса | Через имя класса | Напрямую через имя метода |  |
|------------------------|------------------|---------------------------|--|

**10. Если в описании класса явно не указан спецификатор доступа, то по-умолчанию он**

|         |           |        |  |
|---------|-----------|--------|--|
| Private | Protected | Public |  |
|---------|-----------|--------|--|

**11. Когда при использовании ссылки может возникнуть анонимный объект?**

|   |                                 |  |  |
|---|---------------------------------|--|--|
| Когда тип ссылки не совпадает с типом объекта | Когда устанавливается ссылка на |  |  |
|---|---------------------------------|--|--|

|  |                     |  |  |
|--|---------------------|--|--|
|  | динамический объект |  |  |
|--|---------------------|--|--|

**12.** В чем особенность использования параметров функции по-умолчанию?

|                  |  |  |  |
|------------------|--|--|--|
| Нет особенностей | Параметры должны располагаться слева направо | Параметры должны располагаться справа налево |  |
|------------------|--|--|--|

Правильные ответы: 3

**13.** Выберите правильные утверждения

|   |   |   |  |
|---|---|---|--|
| Метод класса, перегружающий бинарную операцию, должен иметь два параметра | Метод класса, перегружающий бинарную операцию, должен иметь один параметр | Метод класса, перегружающий бинарную операцию, не должен иметь параметров |  |
|---|---|---|--|

***Критерий оценивания результатов тестирования***

| <b><i>Баллы, %</i></b> | <b><i>Оценка</i></b> |
|------------------------|----------------------|
| 99-100                 | Превосходно          |
| 91-98                  | Отлично              |
| 86-90                  | Очень хорошо         |
| 71-85                  | Хорошо               |
| 51-70                  | Удовлетворительно    |
| 31-50                  | Неудовлетворительно  |
| 0-30                   | Плохо                |

**6. Учебно-методическое и информационное обеспечение дисциплины**

а) основная литература:

1. Керниган Б., Ритчи Д. Язык программирования Си. - М. Финансы и статистика, 1992; СПб.: «Невский Диалект», 2001, 2004. – 352с. (16 экз.)
2. Павловская Т.А. C/C++. Программирование на языке высокого уровня. – СПб.: Питер, 2002, 2004, 2007, 2010, 2011, 2016. – 461с. (16 экз.)

б) дополнительная литература:

1. Фомина И.А. Методические указания по курсу "Язык программирования СИ" (разделы: указатели, массивы, функции). - Нижний Новгород: ННГУ им. Н.И. Лобачевского, 2005. - 42 с. [http://www.unn.ru/books/resources.html, рег. номер 958.15.08]

в) программное обеспечение и Интернет-ресурсы

Для обеспечения самостоятельной работы обучающихся используется электронный курс «Языки и методы программирования (Гришагин В.А.)»

<https://e-learning.unn.ru/enrol/index.php?id=6055>

созданные в системе электронного обучения ННГУ – <https://e-learning.unn.ru/>.

## **7. Материально-техническое обеспечение дисциплины**

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения:

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду

Программа составлена в соответствии с требованиями ФГОС ВО (3++) по направлению подготовки 02.03.01 Математика и компьютерные науки.

Автор (ы): Штанюк А.А.

Заведующий кафедрой: Прилуцкий М.Х.

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики от 1 декабря 2021 года, протокол № 2.