

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 совет ННГУ  
протокол от  
«14» декабря 2021 г. № 4

**Рабочая программа дисциплины**

**Алгоритмы и структуры данных**

*(наименование дисциплины (модуля))*

Уровень высшего образования

бакалавриат

*(бакалавриат / магистратура / специалитет)*

Направление подготовки / специальность

**01.03.02 Прикладная математика и информатика**

*(указывается код и наименование направления подготовки / специальности)*

Направленность образовательной программы

**Системный анализ, исследование операций и управление**

*(указывается профиль / магистерская программа / специализация)*

Форма обучения

**очно-заочная**

*(очная / очно-заочная / заочная)*

Нижегород

2022 год

### 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.О.11, “Алгоритмы и структуры” данных относится к обязательной части ООП направления подготовки 01.03.02 Прикладная математика и информатика.

### 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**  |                                     |
| <i>ОПК-2: Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач</i> | <i>ОПК-2.1.: Знает математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач</i>  | <b>ЗНАТЬ:</b><br><i>основные алгоритмические конструкции и программные решения основы построения математических, имитационных и информационных моделей</i>   | <i>Собеседование, тест</i>          |
|  | <i>ОПК-2.2.: Умеет осуществлять выбор и адаптацию математических методов и программного обеспечения для разработки и реализации алгоритмов решения задач в области профессиональной деятельности</i> | <b>УМЕТЬ:</b><br><i>- разрабатывать программные решения на базе математических, имитационных и информационных моделей,<br/>- создавать ресурсы для глобальной сети, БД,<br/>- создавать тесты для тестирования систем на соответствие требованиям и стандартам</i> | <i>Проект (лабораторная работа)</i> |

### 3. Структура и содержание дисциплины

#### 3.1. Трудоемкость дисциплины

|  | Очная форма обучения |
|--|----------------------|
| <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>116</b>           |
| - занятия лекционного типа                     | <b>64</b>            |
| - занятия семинарского типа                    | <b>32</b>            |
| - занятия лабораторного типа                   | <b>16</b>            |
| - текущий контроль (КСР)                       |                      |

|   |     |
|---|-----|
|   | 4   |
| самостоятельная работа                    | 208 |
| Промежуточная аттестация – экзамен, зачет | 36  |
| в том числе (3 семестр)                   |     |
| Общая трудоемкость                        | 5   |
| Часов по учебному плану                   | 180 |
| аудиторные занятия (контактная работа):   | 49  |
| - занятия лекционного типа                | 32  |
| - занятия семинарского типа               | 16  |
| - занятия лабораторного типа              |     |
| - текущий контроль (КСР)                  | 1   |
| самостоятельная работа                    | 131 |
| Промежуточная аттестация – зачет          |     |
| в том числе (4 семестр)                   |     |
| Общая трудоемкость                        | 5   |
| Часов по учебному плану                   | 180 |
| аудиторные занятия (контактная работа):   | 67  |
| - занятия лекционного типа                | 32  |
| - занятия семинарского типа               | 16  |
| - занятия лабораторного типа              | 16  |
| - текущий контроль (КСР)                  | 3   |
| самостоятельная работа                    | 77  |
| Промежуточная аттестация – экзамен/зачет  | 36  |

### 3.2. Содержание дисциплины

| Наименование и краткое содержание разделов и тем дисциплины | Всего (часы) | В том числе   |                           |                            |       | Самостоятельная работа обучающегося, часы |
|---|--------------|---|---------------------------|----------------------------|-------|---|
|   |              | Контактная работа (работа во взаимодействии с преподавателем), часы. Из них |                           |                            |       |   |
|   |              | Занятия лекционного типа  | Занятия семинарского типа | Занятия лабораторного типа | Всего |   |
| <b>Семестр 3</b>  |              |   |                           |                            |       |   |
| Технологии программирования. Базовые концепции ООП.         | 17           | 6   |                           |                            | 6     | 11  |
| Классы С++. Стандартные члены класса.                       | 40           | 6   | 4                         |                            | 10    | 30  |
| Перегрузка операций.  | 40           | 6   | 4                         |                            | 10    | 30  |
| Наследование и полиморфизм                                  | 42           | 8   | 4                         |                            | 12    | 30  |

|   |            |           |           |           |            |            |
|---|------------|-----------|-----------|-----------|------------|------------|
| Шаблоны функций и классов   | 40         | 6         | 4         |           | 10         | 30         |
| Текущий контроль (КСР)  | 1          |           |           |           | 1          |            |
| Промежуточная аттестация – зачет  |            |           |           |           |            |            |
| <b>Итого</b>  | <b>180</b> | <b>32</b> | <b>16</b> |           | <b>49</b>  | <b>131</b> |
| <b>Семестр 4</b>  |            |           |           |           |            |            |
| Понятие структуры данных. Схема и экземпляр СД.<br>Понятие структуры хранения | 14         | 2         |           |           | 2          | 12         |
| Структура хранения множества  | 22         | 6         | 3         | 3         | 12         | 10         |
| Структура хранения матриц специального вида                                   | 24         | 6         | 3         | 3         | 12         | 12         |
| Динамические структуры данных (Стек, Очередь)                                 | 24         | 6         | 3         | 3         | 12         | 12         |
| Линейные списки   | 28         | 6         | 3         | 3         | 12         | 16         |
| Таблицы   | 29         | 6         | 4         | 4         | 14         | 15         |
| Текущий контроль (КСР)  | 3          |           |           |           | 3          |            |
| Итоговая аттестация – экзамен/зачет   | 36         |           |           |           |            |            |
| Итого   | <b>180</b> | <b>32</b> | <b>16</b> | <b>16</b> | <b>67</b>  | <b>77</b>  |
| <b>Итого</b>  | <b>360</b> | <b>64</b> | <b>32</b> | <b>16</b> | <b>116</b> | <b>208</b> |

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 «Алгоритмы и структуры данных»**

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

Для обеспечения самостоятельной работы обучающихся используется электронный курс, созданный в системе электронного обучения ННГУ - <https://e-learning.unn.ru/>.

<https://e-learning.unn.ru/enrol/index.php?id=254>

#### **Выполнение лабораторных работ на следующие темы:**

Лабораторная работа «Верхнетреугольные матрицы на шаблонах»

Лабораторная работа «Реализация класса Стек на списке»

Лабораторная работа «Реализация класса Очередь на динамическом массиве»

Лабораторная работа «Представление полиномов на основе списка»

Лабораторная работа «Таблица на сбалансированных деревьях»

Лабораторная работа «Реализация хеш-таблицы»

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   | При решении стандартных задач не продемонстрированы                    | Продемонстрированы основные умения. Решены                            | Продемонстрированы все основные умения. Решены все  | Продемонстрированы все основные умения. Решены все  | Продемонстрированы все основные умения, решены все                         | Продемонстрированы все основные умения, решены все         |

|               |  |   |  |  |   |  |   |
|---------------|--|---|--|--|---|--|---|
|               | оценить наличие умений вследствие отказа обучающегося от ответа  | основные умения.<br><br>Имели место грубые ошибки.  | типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.       | основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами. | основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.     | основные задачи с отдельными несущественным недочетами, выполнены все задания в полном объеме. | 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. | Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами                  | Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов. | Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов.              | Продемонстрирован 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 (3 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 <code>new</code> и <code>delete</code>. 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 <code>std</code>.</li> <li>20. Перегрузка операторов.</li> <li>21. Правила перегрузки операторов.</li> <li>22. Программа моделирования комплексных чисел.</li> <li>23. Проблемы при копировании объектов, содержащих указатели.</li> <li>24. Конструктор копирования.</li> <li>25. Вывод в языке C++.</li> <li>26. Ввод в языке C++.</li> <li>27. Перегрузка операторов ввода для пользовательских типов.</li> <li>28. Перегрузка операторов вывода для пользовательских типов.</li> <li>29. Файловый ввод и вывод.</li> </ol> | <p><i>ОПК-2</i></p> |
| <ol style="list-style-type: none"> <li>30. Ссылки. Ссылка как аргумент функции. Возврат ссылки из функции.</li> <li>31. Операторы <code>new</code> и <code>delete</code>. Привести примеры использования.</li> <li>32. Функции как члены структуры.</li> <li>33. Встроенные функции.</li> <li>34. Классы. Инкапсуляция. Члены класса. Скрытие данных. Объекты класса.</li> <li>35. Конструкторы.</li> <li>36. Размер класса и размер объектов класса.</li> <li>37. Друзья класса. Привести пример.</li> <li>38. Копирование объектов класса.</li> <li>39. Структуры и классы. Правила доступа к членам класса.</li> <li>40. Статические члены класса. Привести пример.</li> </ol>   | <p><i>ОПК-2</i></p> |

41. Ссылка на себя.
42. Деструкторы. Деструктор в классе дат.
43. Область действия имен. Глобальные и локальные имена. Статические имена.
44. Заголовочные файлы. Страж включения.
45. Понятие стека. Реализация стека в виде массива.
46. Программа анализа правильности расстановки скобок в тексте.
47. Пространства имен.
48. Стандартное пространство имен std.
49. Перегрузка операторов.
50. Правила перегрузки операторов.
51. Программа моделирования комплексных чисел.
52. Проблемы при копировании объектов, содержащих указатели.
53. Конструктор копирования.
54. Вывод в языке C++.
55. Ввод в языке C++.
56. Перегрузка операторов ввода для пользовательских типов.
57. Перегрузка операторов вывода для пользовательских типов.
58. Файловый ввод и вывод.
59. Сравнение текстового и бинарного файлов.
60. Объекты как члены класса.
61. Конструкторы встроенных типов.
62. Наследование классов.
63. [Управление доступом при наследовании.](#)
64. **Ошибка! Закладка не определена.**
65. Производный класс личных данных.
66. Виртуальные функции.
67. Абстрактные классы
68. Вызов виртуальных функций.
69. Локальные классы.
70. Совместимость типов при наследовании.
71. Множественное наследование.
72. Класс алгебраических векторов Vector.
73. Класс прямоугольных матриц.
74. Класс систем линейных уравнений.
75. Шаблоны функций.
76. Шаблоны классов.
77. Шаблон классов векторов.
78. Шаблон классов динамических массивов.
79. Обработка исключений.
80. Стандартная библиотека шаблонов. Привести пример использования шаблонов стандартной библиотеки.
81. Динамическое создание объектов класса. Виртуальные

### Вопросы к экзамену (4 семестр):

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 ++ с и с ++ 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.

Вопрос 2.1. Понятие структуры данных (СД). Схема и экземпляр СД. Примеры: вектор, матрица. Понятие структуры хранения. Примеры.

Вопрос 2.2. Тип данных: список, реализованный на статическом массиве

Вопрос 2.3. Тип данных: стек, реализованный на статическом массиве

Вопрос 2.4. Тип данных: шаблон стека, реализованный на статическом массиве

- Вопрос 2.5. Тип данных: стек, реализованный на динамическом массиве  
 Вопрос 2.7. Тип данных: шаблон класса очередь с круговым заполнением  
 Вопрос 2.8. Тип данных: односвязный список  
 Вопрос 2.9. Тип данных: бинарное дерево поиска  
 Вопрос 2.10. Тип данных: AVL-дерево  
 Вопрос 2.11. Тип данных: красно-черное дерево  
 Вопрос 2.12. Тип данных: хэш-таблица

### 5.2.2. Типовые тестовые задания для оценки сформированности компетенции \_\_ОПК-2\_\_

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>Баллы, %</b> | <b>Оценка</b>       |
|-----------------|---------------------|
| 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

Национальный исследовательский Нижегородский государственный университет  
им. Н.И. Лобачевского  
Институт ИИТММ  
Кафедра Программная инженерия  
Дисциплина Алгоритмы и структуры данных

#### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Технологии программирования: структурное, модульное, ООП. Общее описание.
2. Понятие структуры данных (СД). Схема и экземпляр СД. Примеры: вектор, матрица. Понятие структуры хранения. Примеры.

Зав. кафедрой \_\_\_\_\_  
Экзаменатор \_\_\_\_\_

### 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. С/С++. 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

- Microsoft Visual Studio 2010-2015

### 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 Образовательного стандарта ННГУ по направлению подготовки 01.03.02 Прикладная математика и информатика

Автор (ы) \_\_\_\_\_ Штанюк А.А.

Рецензент (ы) \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_ Зорин А.М.

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики от 01.12.2021 года, протокол № 2.