МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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)

|  |
| --- |
| УТВЕРЖДАЮ: |

|  |  |  |
| --- | --- | --- |
|  Директор |  | В.П.Гергель |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| « |  | » |  |  | 201\_ г. |

**Рабочая программа дисциплины**

|  |
| --- |
| Языки и методы программирования |

(наименование дисциплины (модуля))

Уровень высшего образования

|  |
| --- |
| бакалавриат |

(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность

|  |
| --- |
| 01.03.01 Математика |

 (указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы

|  |
| --- |
| Математика (общий профиль) |

(указывается профиль / магистерская программа / специализация)

Квалификация (степень)

|  |
| --- |
| бакалавр |

(бакалавр / магистр / специалист)

Форма обучения

|  |
| --- |
| Очная |

 (очная / очно-заочная / заочная)

Нижний Новгород

2017

1. **Место и цели дисциплины (модуля) в структуре ОПОП**

Дисциплина «Языки и методы программирования» относится к базовой части ОПОП (Б1.Б20 - базовая часть) и обязательна для освоения на 1-2 годах обучения (2,3 семестр). Материал дисциплины опирается на знание курсов «Основы информатики» (1 семестр).

**Целями освоения дисциплины являются**:

*Целями освоения дисциплины являются:*

* углубление знаний в области языков программирования С/С++
* знакомство с основами объектно-ориентированного программирования на языке С++
* знакомство со стандартной библиотекой шаблонов STL
* изучение базовых структур данных и алгоритмов
1. **Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников)**

|  |  |
| --- | --- |
| **Формируемые компетенции**(код компетенции, уровеньосвоения – при наличии в карте компетенции) | **Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций** |
| ОК-7 Способность к самоорганизации и самообразованиюНачальный этап. | УметьУ(ОК-7) Самостоятельно решать задачи по составлению компьютерных программВладетьВ(ОК-7) Навыками по планированию времени при решении алгоритмических задач |
| ОПК-2 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасностиНачальный этап. | ЗНАТЬЗ(ОПК2) Основные источники и методы поиска научной информациии общечеловеческой культуры.УМЕТЬУ1(ОПК2) Осуществлять подбор литературы и поиск нужной информации в сети интернетУ2(ОПК2) Находить и выбирать наиболее эффективные методы решения алгоритмических задачВладеть:В1(ОПК2) Методикой выбора решения стандартных задач с применением стандартных алгоритмов |
| ОПК-4Способность находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных средствНачальный этап. | Знать:З(ОПК-4) Основы синтаксиса языков программирования С/С++Уметь: У1(ОПК-4) Осуществлять оценку сложности алгоритмов для решения задачУ2(ОПК-4) Писать программы на языках С/С++ для решения задачВладеть: В(ОПК4)современными инструментальными вычислительными средствами. |
| ПК-4способностью публично представлять собственные и известные научные результатыНачальный этап. | Знать:З(ПК-4)Основы оформления текстов программУметь:У1(ПК-4) Подготавливать научно-техническую информацию для представления своих программ, схем алгоритмов |

1. **Структура и содержание дисциплины Языки и методы программирования**

Объем дисциплины (модуля) составляет 8 зачетных единиц, всего 288 часов, из которых 163 часа составляет контактная работа обучающегося с преподавателем (64 часов занятия лекционного типа, 96 часов занятия семинарского типа, 3 часа промежуточной аттестации), 125 часа составляет самостоятельная работа обучающегося (в т.ч. 36 часов подготовки к экзамену).

|  |  |  |
| --- | --- | --- |
| **Наименование и краткое содержание разделов и тем дисциплины,** **форма промежуточной аттестации по дисциплине** | **Всего****(часы)** | в том числе |
| **контактная работа (работа во взаимодействии с преподавателем), часы**из них | **Самостоятельная** **работа студента** **часы** |
|  **Занятия лекционного типа** |  **Занятия семинарского типа** |  **Лабораторные работы** |  | **Всего контактных часов** | **СРС** |  | **Всего СРС** |
| Технологии программирования. Базовые концепции ООП. | 20 | 8 | 8 |  |  | 16 | 4 |  | 4 |
| Классы С++. Стандартные члены класса. | 22 | 6 | 6 |  |  | 12 | 10 |  | 10 |
| Агрегация | 22 | 6 | 6 |  |  | 12 | 10 |  | 10 |
| Разработка простейших структур данных | 22 | 6 | 6 |  |  | 12 | 10 |  | 10 |
| Перегрузка операций. | 22 | 6 | 6 |  |  | 12 | 10 |  | 10 |
| В т.ч. текущий контроль |  |  |  |  |  |  |  |  |  |
| **Промежуточная аттестация - Зачет**. |
| Связанные списки | 44 | 8 | 8 | 8 |  | 24 | 20 |  | 20 |
| Бинарные деревья | 38 | 6 | 6 | 6 |  | 18 | 20 |  | 20 |
| Наследование и полиморфизм | 38 | 6 | 6 | 6 |  | 18 | 20 |  | 20 |
| Шаблоны функций и классов | 32 | 6 | 9 | 6 |  | 21 | 11 |  | 11 |
| Стандартная библиотека шаблонов | 28 | 6 | 6 | 6 |  | 18 | 10 |  | 10 |
| В т.ч. текущий контроль | 3 |  | 3 |  |  |  |  |  |  |
| **Итоговая аттестация – зачет, экзамен** |

1. **Образовательные технологии**

Используются 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 материла 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 трояким сочетанием: изложение новой учебной информации лектором, постановка вопросов и организация дискуссии в поиске ответов на поставленные вопросы». Поводится 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.

Для выполнения лабораторных работы предусматривается работа в терминал-классах.

1. **Учебно-методическое обеспечение самостоятельной работы обучающихся**

А. Виды самостоятельной работы студентов

- проработка тем для самостоятельной подготовки

- подготовка к промежуточной и итоговой аттестациям

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.

1. ***Особенности функций C++***

Перегрузки функций. Аргументы функций по умолчанию.

1. ***Ссылки. Работа с памятью***

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

1. ***Структуры в C++***

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

1. ***Понятие класса***

Скрытие данных. Открытые и закрытые члены.

1. ***Конструкторы класса***

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

1. ***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.

1. ***Копирование объектов класса***

Встроенный механизм копирования.

1. ***Друзья класса***

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

1. ***Управление доступом***

Структуры и классы. Правила доступа.

1. ***Деструкторы***

Синтаксис. Назначение. Использование.

1. ***Программы из нескольких файлов***

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

1. ***Пространство имен***

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

1. ***Перегрузка операторов***

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

1. ***Конструктор копирования и оператор присваивания***

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

1. ***Ввод и вывод***

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

1. ***Ввод и вывод пользовательских типов данных***

Перегрузка операторов ввода и вывода для пользовательских типов данных.

1. ***Объекты как члены класса***

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

1. ***Наследование***

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

1. ***Полиморфизм***

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

1. ***Абстрактные классы***

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

1. ***Множественное наследование***

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

1. ***Шаблоны***

Шаблоны функций. Шаблоны классов.

1. ***Стандартная библиотека шаблонов***

Контейнеры. Алгоритмы. Итераторы.

1. ***Обработка исключений***

Понятие исключения. Синтаксис обработки исключений.

**Список вопросов к зачету (2 семестр)**

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 (3 семестр)**

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.
53. **Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю**)

**6.1 Перечень компетенций выпускников образовательной программы с указанием результатов обучения**

|  |  |  |
| --- | --- | --- |
| Индикаторы компетенции | Критерии оценивания (дескрипторы) | Шкала оценивания |
| ***Уметь******У(ОК-7) Самостоятельно решать задачи по составлению компьютерных программ******Владеть******В(ОК-7) Навыками по планированию времени при решении алгоритмических задач*** | Отсутствие знаний материала, отсутствует способность решения стандартных задач, полное отсутствие навыков, предусмотренных компетенцией. | Плохой уровень формирования компетенции. «Плохо» |
| Наличие грубых ошибок в основном материале,наличие грубых ошибок при решении стандартных задач,отсутствие навыков, предусмотренных данной компетенцией | Неудовлетворительный уровень формирования компетенции. «неудовлетворительно» |
| **Знать** некоторые основные понятия и свойства изучаемых алгоритмов **Уметь**У1 с погрешностями.**Уметь**У2 с погрешностями. **Владеть** некоторыми основными навыками, демонстрируя их в стандартных ситуациях | Удовлетворительный уровень формирования компетенции. «Удовлетворительно» |
| **Знать** большинство основных понятий и свойств изучаемых последовательных алгоритмов и их параллельных обобщений. **Уметь**У1 с незначительными погрешностями.**Уметь**У2 с незначительными погрешностями. **Владеть** основными навыками, демонстрируя их в стандартных ситуациях | Хороший уровеньформирования компетенции. «Хорошо» |
| **Знать** алгоритмы и структуры данных. **Уметь**У1 с незначительными погрешностями. Уметь У2 с незначительными погрешностями **Владеть** всеми основными навыками, демонстрируя их в стандартных ситуациях | Очень хороший уровеньформирования компетенции  «Очень хорошо» |
| **Знать** основные методы и алгоритмы, предусмотренные компетенцией без ошибок и погрешностей. **Уметь** У1 в полном объеме. Уметь У2 в полном объеме. **Владеть** всеми навыками, демонстрируя их в стандартных ситуациях. | Отличный уровеньформирования компетенции  «Отлично» |
| **Знать** основной и дополнительный материал без ошибок и погрешностей. **Уметь**У1 в полном объеме. Уметь У2 в полном объеме. Свободно **Владеть** всеми навыками, демонстрируя их в стандартных и нестандартных ситуациях. | Превосходный уровеньформирования компетенции  «Превосходно» |

|  |  |  |
| --- | --- | --- |
| Индикаторы компетенции | Критерии оценивания (дескрипторы) | Шкала оценивания |
| ***ЗНАТЬ****З(ОПК-2) Основные источники и методы поиска научной информации и общечеловеческой культуры.****УМЕТЬ****У1(ОПК-2) Осуществлять подбор литературы и поиск нужной информации в сети интернет**У2(ОПК-2) Находить и выбирать наиболее эффективные методы решения алгоритмических задач****Владеть:****В1(ОПК-2) Методикой выбора решения стандартных задач с применением стандартных алгоритмов* | Отсутствие знаний материала, отсутствует способность решения стандартных задач, полное отсутствие навыков, предусмотренных компетенцией. | Плохой уровеньформирования компетенции. «Плохо» |
| Наличие грубых ошибок в основном материале, наличие грубых ошибок при решении стандартных задач, отсутствие навыков, предусмотренных данной компетенцией | Неудовлетворительный уровень формирования компетенции. «неудовлетворительно» |
| **Знать** некоторые основные понятия и свойства изучаемых алгоритмов **Уметь**У1 с погрешностями.**Уметь**У2 с погрешностями. **Владеть** некоторыми основными навыками, демонстрируя их в стандартных ситуациях | Удовлетворительный уровень формирования компетенции. «Удовлетворительно» |
| **Знать** большинство основных понятий и свойств изучаемых последовательных алгоритмов и их параллельных обобщений. **Уметь**У1 с незначительными погрешностями.**Уметь**У2 с незначительными погрешностями. **Владеть** основными навыками, демонстрируя их в стандартных ситуациях | Хороший уровеньформирования компетенции. «Хорошо» |
| **Знать** алгоритмы и структуры данных. **Уметь**У1 с незначительными погрешностями. Уметь У2 с незначительными погрешностями **Владеть** всеми основными навыками, демонстрируя их в стандартных ситуациях | Очень хороший уровеньформирования компетенции  «Очень хорошо» |
| **Знать** основные методы и алгоритмы, предусмотренные компетенцией без ошибок и погрешностей. **Уметь** У1 в полном объеме. Уметь У2 в полном объеме.**Владеть** всеми навыками, демонстрируя их в стандартных ситуациях. | Отличный уровеньформирования компетенции  «Отлично» |
| **Знать** основной и дополнительный материал без ошибок и погрешностей. **Уметь**У1 в полном объеме. Уметь У2 в полном объеме. Свободно **Владеть** всеми навыками, демонстрируя их в стандартных и нестандартных ситуациях. | Превосходный уровеньформирования компетенции  «Превосходно» |

|  |  |  |
| --- | --- | --- |
| Индикаторы компетенции | Критерии оценивания (дескрипторы) | Шкала оценивания |
| Знать:*З(ОПК-4) Основы синтаксиса языков программирования С/С++****Уметь:*** *У1(ОПК-4) Осуществлять оценку сложности алгоритмов для решения задач**У2(ОПК-4) Писать программы на языках С/С++ для решения задач****Владеть: В1(ОПК-4)****современными инструментальными вычислительными средствами.* | Отсутствие знаний материала, отсутствует способность решения стандартных задач, полное отсутствие навыков, предусмотренных компетенцией. | Плохой уровеньформирования компетенции. «Плохо» |
| Наличие грубых ошибок в основном материале, наличие грубых ошибок при решении стандартных задач, отсутствие навыков, предусмотренных данной компетенцией | Неудовлетворительный уровень формирования компетенции. «неудовлетворительно» |
| **Знать** некоторые основные понятия и свойства изучаемых алгоритмов **Уметь**У1 с погрешностями.**Уметь**У2 с погрешностями. **Владеть** некоторыми основными навыками, демонстрируя их в стандартных ситуациях | Удовлетворительный уровень формирования компетенции. «Удовлетворительно» |
| **Знать** большинство основных понятий и свойств изучаемых последовательных алгоритмов и их параллельных обобщений. **Уметь**У1 с незначительными погрешностями.**Уметь**У2 с незначительными погрешностями. **Владеть** основными навыками, демонстрируя их в стандартных ситуациях | Хороший уровеньформирования компетенции. «Хорошо» |
| **Знать** алгоритмы и структуры данных. **Уметь**У1 с незначительными погрешностями. Уметь У2 с незначительными погрешностями **Владеть** всеми основными навыками, демонстрируя их в стандартных ситуациях | Очень хороший уровеньформирования компетенции  «Очень хорошо» |
| **Знать** основные методы и алгоритмы, предусмотренные компетенцией без ошибок и погрешностей. **Уметь** У1 в полном объеме. Уметь У2 в полном объеме.**Владеть** всеми навыками, демонстрируя их в стандартных ситуациях. | Отличный уровеньформирования компетенции  «Отлично» |
| **Знать** основной и дополнительный материал без ошибок и погрешностей. **Уметь**У1 в полном объеме. Уметь У2 в полном объеме. Свободно **Владеть** всеми навыками, демонстрируя их в стандартных и нестандартных ситуациях. | Превосходный уровеньформирования компетенции  «Превосходно» |

|  |  |  |
| --- | --- | --- |
| Индикаторы компетенции | Критерии оценивания (дескрипторы) | Шкала оценивания |
| Знать:*З(ПК4) Основы оформления текстов программ*Уметь:*У1(ПК4) Подготавливать научно-техническую информацию для представления своих программ, схем алгоритмов* | Отсутствие знаний материала, отсутствует способность решения стандартных задач, полное отсутствие навыков, предусмотренных компетенцией. | Плохой уровеньформирования компетенции. «Плохо» |
| Наличие грубых ошибок в основном материале, наличие грубых ошибок при решении стандартных задач, отсутствие навыков, предусмотренных данной компетенцией | Неудовлетворительный уровень формирования компетенции. «неудовлетворительно» |
| **Знать** некоторые основные понятия и свойства изучаемых алгоритмов **Уметь**У1 с погрешностями.**Уметь**У2 с погрешностями. **Владеть** некоторыми основными навыками, демонстрируя их в стандартных ситуациях | Удовлетворительный уровень формирования компетенции. «Удовлетворительно» |
| **Знать** большинство основных понятий и свойств изучаемых последовательных алгоритмов и их параллельных обобщений. **Уметь**У1 с незначительными погрешностями.**Уметь**У2 с незначительными погрешностями. **Владеть** основными навыками, демонстрируя их в стандартных ситуациях | Хороший уровеньформирования компетенции. «Хорошо» |
| **Знать** алгоритмы и структуры данных. **Уметь**У1 с незначительными погрешностями. Уметь У2 с незначительными погрешностями **Владеть** всеми основными навыками, демонстрируя их в стандартных ситуациях | Очень хороший уровеньформирования компетенции  «Очень хорошо» |
| **Знать** основные методы и алгоритмы, предусмотренные компетенцией без ошибок и погрешностей. **Уметь** У1 в полном объеме. Уметь У2 в полном объеме. **Владеть** всеми навыками, демонстрируя их в стандартных ситуациях. | Отличный уровеньформирования компетенции  «Отлично» |
| **Знать** основной и дополнительный материал без ошибок и погрешностей. **Уметь**У1 в полном объеме. Уметь У2 в полном объеме. Свободно **Владеть** всеми навыками, демонстрируя их в стандартных и нестандартных ситуациях. | Превосходный уровеньформирования компетенции  «Превосходно» |

**Карта компетенций для оценивания умений и навыков**

|  |  |
| --- | --- |
| Индикаторыкомпетенции | Критерии оценивания (дескрипторы) |
| «плохо» | «неудовлетворительно» | «удовлетворительно» | «хорошо» | «очень хорошо» | «отлично» | «превосходно» |
| УменияУ(ОК7),У1(ОПК2),У2(ОПК2)У1(ОПК4), У2(ОПК4),У(ПК4) | отсутствует способность решения стандартных задач | наличие грубых ошибок при решении стандартных задач | способность решения основных стандартных задач с негрубыми ошибками | способность решения всех стандартных задач с незначительными погрешностями | способность решения всех стандартных задач без ошибок и погрешностей | Способность решения стандартных и некоторых нестандартных задач | способность решения стандартных задач и широкого круга нестандартных задач |
| НавыкиВ1(ОК7),В(ОПК4),В(ОПК4) | полное отсутствие навыков, предусмотренных компетенцией | отсутствие ряда важнейших навыков, предусмотренных данной компетенцией | наличие минимально необходимого множества навыков | наличие большинства основных навыков, продемонстрированное в стандартных ситуациях | наличие всех основных навыков, продемонстрированных в стандартных ситуациях | наличие всех навыков, продемонстрированное в стандартных ситуациях | Наличие всех навыков, продемонстрированное в стандартных и нестандартных ситуациях |
| Личностные качестваОК7 | соответствующие личностные качетсва не сформированы | сформированностьличностных качеств недостаточный для достижения основных целей обучения | сформированностьличностных качеств минимально необходимая для достижения основных целей обучения | личностные качества в целом сформирваны | сформированные личностные качества достаточны для достижения целей обучения | Личностные качества сформированы на высоком уровне | Сформированность личностных качестввыше обязательных требований |

**6.2 Описание шкал оценивания**

Для оценивания результатов учебной деятельности студентов при изучении дисциплины «Языки и методы программирования» используется балльная система оценки учебной работы студентов. По результатам промежуточной аттестации проставляются оценки «Зачтено» (соответствует уровням оценки компетенций «удовлетворительно» и выше) и «Не зачтено» (соответствует уровням оценки компетенций «плохо» и «неудовлетворительно»). По результатам итоговой аттестации проставляются оценки «Превосходно», «Отлично», «Очень хорошо», «Хорошо», «Удовлетворительно» и «Неудовлетворительно» в соответствии со шкалой оценивания в таблице компетенций.

|  |  |
| --- | --- |
| Требования к результатам освоения дисциплины | Зачет |
| Знание основных вопросов тем дисциплины, умение использовать их в практических ситуациях при написании программного кода. | Зачтено |
| Отсутствие у обучаемого комплекса знаний по темам дисциплины, отсутствие практических навыков | Не зачтено |

**6.3 Критерии и процедуры оценивания результатов обучения по дисциплине (модулю), характеризующих этапы формирования компетенций ОК-7, ОПК-2, ОПК-4, ПК-4.**

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.

**6.4 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения, характеризующих этапы формирования компетенций и (или) для итогового контроля сформированности компетенций ОК-7, ОПК-2, ОПК-4, ПК-4.**

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 программирова-ния C++. Примеры.

Вопрос 1.4. Концепции ООП: полиморфизм. Поддержка в языке программирова-ния C++. Примеры.

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

Вопрос 1.6. Создание объектов. Конструкторы. Виды конструкторов, объявление. Реализация конструктов 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 использова-ния.

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. Друже-ственные 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.

**Примеры тестов для оценивания результатов формирования знаний, умений ОК-7, ОПК-2, ОПК-4, ПК-4.**

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 С++ 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***

|  |  |
| --- | --- |
| ***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-2, ОПК-4, ПК-4

Национальный исследовательский Нижегородский

государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Кафедра программной инженерии

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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6.5. Методические материалы, определяющие процедуры оценивания**

Положение о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся в ННГУ от 13.02.2014. [http://www.unn.ru/site/images/docs/obrazovorg/Formi\_stroki\_kontrolya\_13.02.2014.pdf](http://www.unn.ru/site/images/docs/obrazov-org/Formi_stroki_kontrolya_13.02.2014.pdf)

**7. Учебно-методическое и информационное обеспечение дисциплины (модуля)**

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

* Microsoft Visual Studio 2010-2015

**8. Материально-техническое обеспечение дисциплины (модуля)**

Имеются в наличии учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля, промежуточной аттестации, а также помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет».

Программа составлена в соответствии с требованиями ФГОС ВО с учетом рекомендаций ОПОП ВО по направлению (01.03.01), специальности (специализации) Математика.

Автор (ы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Штанюк А.А.

Рецензент (ы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Заведующий кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гергель В.П

Программа одобрена на заседании методической комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ института ИТММ

от \_\_\_\_\_\_\_\_\_\_\_ года, протокол № \_\_\_\_\_\_\_\_.