

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

Уровень высшего образования
бакалавриат

Направление подготовки
02.03.01 Математика и компьютерные науки

Направленность образовательной программы
Общий профиль

Квалификация (степень)
бакалавр

Форма обучения
Очная

Нижний Новгород

2017

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

Дисциплина «Языки и методы программирования» относится к базовой части ОПОП и обязательна для освоения на 1-2 годах обучения (2,3 семестр).

Материал дисциплины опирается на знание курса «Основы информатики».

Целями освоения дисциплины являются:

- углубление знаний в области языков программирования C/C++
- знакомство с основами объектно-ориентированного программирования на языке C++
- знакомство со стандартной библиотекой шаблонов STL
- изучение базовых структур данных и алгоритмов

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников)

Формируемые компетенции (код компетенции, уровень освоения – при наличии в карте компетенции)	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
ОК-7 Способность к самоорганизации и самообразованию	Уметь У(ОК-7) Самостоятельно решать задачи по составлению компьютерных программ Владеть В(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 безопасности	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-
ОПК-4 Способность находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных	ЗНАТЬ: З(ОПК-4) Основы синтаксиса языков программирования C/C++ УМЕТЬ: У1(ОПК-4) Осуществлять оценку сложности алгоритмов для решения задач У2(ОПК-4) Писать программы на языках C/C++ для

<i>вычислительных средств</i>	<i>решения задач</i> ВЛАДЕТЬ: В(ОПК-4) современными инструментальными вычислительными средствами.
ПК-4 <i>способность публично представлять собственные и известные научные результаты</i>	ЗНАТЬ: З(ПК-4) Основы оформления текстов программ УМЕТЬ: У1(ПК-4) Подготавливать научно-техническую информацию для представления своих программ, схем алгоритмов

3. Структура и содержание дисциплины «Языки и методы программирования»

Объем дисциплины составляет 9 зачетных единиц, всего 324 часов, из которых 180 часов составляет контактная работа обучающегося с преподавателем (64 часа занятия лекционного типа, 64 часа занятия семинарского типа, 48 часов занятия лабораторного типа, 4 часа мероприятия промежуточной аттестации), 144 часов составляет самостоятельная работа обучающегося (в т.ч. включая 36 часов подготовки к экзамену).

Наименование и краткое содержание разделов и тем дисциплины, форма промежуточной аттестации по дисциплине	семестр	Всего (часы)	в том числе					Самостоятельная работа студента часы
			контактная работа (работа во взаимодействии с преподавателем), часы					
			Занятия лекционного типа	Занятия семинарского типа	Лабораторные работы	Консультации индивидуальные	Всего контактных часов	
Технологии программирования. Базовые концепции ООП.	2	28	8	8			16	12
Классы C++. Стандартные члены класса.	2	29	6	6	4		16	13
Агрегация	2	29	6	6	4		16	13
Разработка простейших структур данных	2	29	6	6	4		16	13
Перегрузка операций.	2	29	6	6	4		16	13
В т.ч. текущий контроль	2	1						
Промежуточная аттестация: зачет.	2							
Связанные списки	3	49	8	8	8		24	25
Бинарные деревья	3	41	6	6	6		18	23
Наследование и полиморфизм	3	41	6	6	6		18	23
Шаблоны функций и классов	3	41	6	6	6		18	23
Стандартная библиотека шаблонов	3	41	6	6	6		18	23
В т.ч. текущий контроль	3	3						
Промежуточная аттестация: зачет, экзамен.								

4. 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 материала слушателями.

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

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

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

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

Лекция-консультация – по типу «**вопросы—ответы—дискуссия**», является тройным сочетанием: изложение новой учебной информации лектором, постановка вопросов и организация дискуссии в поиске ответов на поставленные вопросы». 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 работ предусматривается работа в терминал-классах.

5. Учебно-методическое обеспечение самостоятельной работы обучающихся

А. Виды самостоятельной работы студентов

- проработка тем для самостоятельной подготовки
- подготовка к промежуточной аттестации

В. Образовательные материалы для студентов

- Список тем для самостоятельной проработки

1. Модульное программирование

Программы из нескольких файлов. Проекты. Заголовочные файлы. Страж включения. Организация связи между модулями.

2. Особенности функций C++

Перегрузки функций. Аргументы функций по умолчанию.

3. Ссылки. Работа с памятью

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

4. Структуры в C++

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

5. Понятие класса

Скрытие данных. Открытые и закрытые члены.

6. Конструкторы класса

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

7. Статические члены. Ссылка на себя

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

8. Копирование объектов класса

Встроенный механизм копирования.

9. Друзья класса

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

10. Управление доступом

Структуры и классы. Правила доступа.

11. Деструкторы

Синтаксис. Назначение. Использование.

12. Программы из нескольких файлов

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

13. Пространство имен

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

14. *Перегрузка операторов*

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

15. *Конструктор копирования и оператор присваивания*

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

16. *Ввод и вывод*

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

17. *Ввод и вывод пользовательских типов данных*

Перегрузка операторов ввода и вывода для пользовательских типов данных.

18. *Объекты как члены класса*

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

19. *Наследование*

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

20. *Полиморфизм*

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

21. *Абстрактные классы*

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

22. *Множественное наследование*

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

23. *Шаблоны*

Шаблоны функций. Шаблоны классов.

24. *Стандартная библиотека шаблонов*

Контейнеры. Алгоритмы. Итераторы.

25. *Обработка исключений*

Понятие исключения. Синтаксис обработки исключений.

- Список вопросов к зачету

1. Ссылки. Ссылка как аргумент функции. Возврат ссылки из функции.
2. Операторы new и delete. Привести примеры использования.
3. Функции как члены структуры.
4. Встроенные функции.
5. Классы. Инкапсуляция. Члены класса. Скрытие данных. Объекты класса.
6. Конструкторы.
7. Размер класса и размер объектов класса.
8. Друзья класса. Привести пример.
9. Копирование объектов класса.
10. Структуры и классы. Правила доступа к членам класса.

11. Статические члены класса. Привести пример.
12. Ссылка на себя. Программа модификации дат.
13. Деструкторы. Деструктор в классе дат.
14. Область действия имен. Глобальные и локальные имена. Статические имена.
15. Заголовочные файлы. Страж включения.
16. Понятие стека. Реализация стека в виде массива.
17. Программа анализа правильности расстановки скобок в тексте.
18. Пространства имен.
19. Стандартное пространство имен std.
20. Перегрузка операторов.
21. Правила перегрузки операторов.
22. Программа моделирования комплексных чисел.
23. Проблемы при копировании объектов, содержащих указатели.
24. Конструктор копирования.
25. Вывод в языке C++.
26. Ввод в языке C++.
27. Перегрузка операторов ввода для пользовательских типов.
28. Перегрузка операторов вывода для пользовательских типов.
29. Файловый ввод и вывод.

- Список вопросов к экзамену

1. Ссылки. Ссылка как аргумент функции. Возврат ссылки из функции.
2. Операторы new и delete. Привести примеры использования.
3. Функции как члены структуры. Программа моделирования временем в виде структуры.
4. Встроенные функции.
5. Классы. Инкапсуляция. Члены класса. Скрытие данных. Объекты класса.
6. Конструкторы. Программа «Конструкторы в классе дат».
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. Программа моделирования многоугольников на плоскости.
15. Область действия имен. Глобальные и локальные имена. Статические имена.
16. Заголовочные файлы. Страж включения.
17. Понятие стека. Реализация стека в виде массива.
18. Программа анализа правильности расстановки скобок в тексте.
19. Пространства имен.
20. Стандартное пространство имен std.
21. Перегрузка операторов. Программа моделирования обыкновенных дробей.
22. Правила перегрузки операторов.
23. Программа моделирования комплексных чисел.
24. Проблемы при копировании объектов, содержащих указатели.
25. Конструктор копирования.
26. Вывод в языке C++.
27. Ввод в языке C++.
28. Перегрузка операторов ввода для пользовательских типов.
29. Перегрузка операторов вывода для пользовательских типов.

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

6. Фонд оценочных средств для промежуточной аттестации по дисциплине

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 ЗНАТЬ З(ОПК-2) Основные источники и методы поиска научной информации и общечеловеческой культуры. УМЕТЬ У1(ОПК-2) Осуществлять подбор литературы и поиск нужной	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	Удовлетворительный уровень формирования компетенции.

<p>информации в сети интернет</p> <p>У2(ОПК-2) Находить и выбирать наиболее эффективные методы решения алгоритмических задач</p> <p>Владеть: В1(ОПК-2) Методикой выбора решения стандартных задач с применением стандартных алгоритмов</p> <p>Знать:</p> <p>З(ОПК-4) Основы синтаксиса языков программирования C/C++</p> <p>Уметь: У1(ОПК-4) Осуществлять оценку сложности алгоритмов для решения задач У2(ОПК4) Писать программы на языках C/C++ для решения задач</p> <p>Владеть: В(ОПК-4) современными инструментальными вычислительными средствами. Знать: З(ПК-4) Основы оформления текстов программ</p> <p>Уметь: У1(ПК-4) Подготавливать научно-техническую информацию для представления своих программ, схем алгоритмов</p>	<p>погрешностями. Уметь У2 с погрешностями. Владеть некоторыми основными навыками, демонстрируя их в стандартных ситуациях.</p>	«Удовлетворительно»
	<p>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</p>	Хороший уровень формирования компетенции. «Хорошо»
	<p>Знать алгоритмы и структуры данных. Уметь У1 с незначительными погрешностями. Уметь У2 с незначительными погрешностями Владеть всеми основными навыками, демонстрируя их в стандартных ситуациях</p>	Очень хороший уровень формирования компетенции «Очень хорошо»
	<p>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.</p>	Отличный уровень формирования компетенции «Отлично»
	<p>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.</p>	Превосходный уровень формирования компетенции «Превосходно»

Карта компетенций для оценивания умений и навыков

Индикаторы компетенции	Критерии оценивания (дескрипторы)						
	«плохо»	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«очень хорошо»	«отлично»	«превосходно»
Умения У(ОК-7), У1(ОПК-2), У2(ОПК-2) У1(ОПК-4), У2(ОПК-4), У(ПК-4)	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
Навыки В1(ОК-7), В(ОПК-2), В(ОПК-4)	полное отсутствие навыков, предусмотренных компетенцией	отсутствие ряда важнейших навыков, предусмотренных данной компетенцией	наличие минимального 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 качества ОК-7	соответствующие личностные качества не сформированы	сформированность личностных качеств недостаточная для достижения основных целей обучения	сформированность личностных качеств минимально необходимая для достижения основных целей обучения	личностные качества в целом сформированы	сформированные личностные качества достаточно для достижения целей обучения	Личностные качества сформированы на высоком уровне	Сформированность личностных качеств выше 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 во втором семестре проставляются оценки «Зачтено» (соответствует уровням оценки компетенций «удовлетворительно» и выше) и «Незачтено» (соответствует уровням оценки компетенций «плохо» и «неудовлетворительно»). По результатам промежуточной 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

Критерии выставления оценок за экзамен

Оценка	Уровень подготовленности, характеризуемый оценкой
Превосходно	Превосходная подготовка с очень незначительными погрешностями
Отлично	Подготовка, уровень которой существенно выше среднего с некоторыми ошибками
Очень хорошо	В целом хорошая подготовка с рядом заметных ошибок
Хорошо	Хорошая подготовка, но со значительными ошибками
Удовлетворительно	Подготовка, удовлетворяющая минимальным требованиям
Неудовлетворительно	Необходима дополнительная подготовка для успешного прохождения испытания
Плохо	Подготовка совершенно недостаточна

6.3 Критерии и процедуры оценивания результатов обучения по дисциплине, 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 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.

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
-------	------	-----------------------	--------------

Правильный ответ: 4

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

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

Правильный ответ: 2

3. Для чего используются классы?

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

Правильный ответ: 3

Критерий оценивания результатов тестирования

<i>Баллы, %</i>	<i>Оценка</i>
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

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/pages/general/norm-acts/attest_stud%202014.pdf

Положение о фонде оценочных средств, утвержденное приказом ректора ННГУ от 10.06.2015 №247-ОД.

7. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Окулов, С.М. Программирование в алгоритмах. [Электронный ресурс] — Электрон. дан. — М. : Издательство "Лаборатория знаний", 2017. — 386 с. — Режим доступа: <http://e.lanbook.com/book/94140> — Загл. с экрана.
2. Логинова, Ф.С. Объектно-ориентированные методы программирования. [Электронный ресурс] : учеб. пособие — Электрон. дан. — СПб. : ИЭО СПбУТУиЭ, 2012. — 208 с. — Режим доступа: <http://e.lanbook.com/book/64040> — Загл. с экрана.
3. Конова, Е.А. Алгоритмы и программы. Язык C++. [Электронный ресурс] : учеб. пособие / Е.А. Конова, Г.А. Поллак. — Электрон. дан. — СПб. : Лань, 2017. — 384 с. — Режим доступа: <http://e.lanbook.com/book/90158> — Загл. с экрана.
4. Васильев, А.Н. Самоучитель C++ с примерами и задачами. Книга + виртуальный CD. [Электронный ресурс] : самоучитель — Электрон. дан. —

СПб. : Наука и Техника, 2016. — 480 с. — Режим доступа:

<http://e.lanbook.com/book/74667> — Загл. с экрана.

5. Васильев, А.Н. Объектно-ориентированное программирование на C++. [Электронный ресурс] — Электрон. дан. — СПб.: Наука и Техника, 2016. — 544 с. — Режим доступа: <http://e.lanbook.com/book/90227> — Загл. с экрана.
6. Бабушкина, И.А. Практикум по объектно-ориентированному программированию. [Электронный ресурс] : учеб. пособие / И.А. Бабушкина, С.М. Окулов. — Электрон. дан. — М. : Издательство "Лаборатория знаний", 2015. — 369 с. — Режим доступа: <http://e.lanbook.com/book/66121> — Загл. с экрана.

б) дополнительная литература:

1. Комлев, Н.Ю. Объектно-Ориентированное Программирование. Хорошая книга для хороших людей. [Электронный ресурс] — Электрон. дан. — М. : СОЛОН-Пресс, 2014. — 298 с. — Режим доступа: <http://e.lanbook.com/book/64985> — Загл. с экрана.
2. Дудецкий, В.Н. Объектно-ориентированные языки программирования: учеб. пособие: в 3 ч. Ч. I. [Электронный ресурс] : учеб. пособие — Электрон. дан. — М. : ФЛИНТА, 2015. — 48 с. — Режим доступа: <http://e.lanbook.com/book/74715> — Загл. с экрана.
3. Дудецкий, В.Н. Объектно-ориентированные языки программирования: учеб. пособие: в 3 ч. Ч. II. [Электронный ресурс] : учеб. пособие — Электрон. дан. — М. : ФЛИНТА, 2015. — 40 с. — Режим доступа: <http://e.lanbook.com/book/74780> — Загл. с экрана.
4. Гаврилова, И.В. Разработка приложений. [Электронный ресурс] : учеб. пособие — Электрон. дан. — М. : ФЛИНТА, 2012. — 241 с. — Режим доступа: <http://e.lanbook.com/book/44746> — Загл. с экрана.
5. Бородихин, В.Н. Языки программирования (СИ/СИ++): учебно-методическое пособие. [Электронный ресурс] : учеб.-метод. пособие — Электрон. дан. — Омск : ОмГУ, 2013. — 200 с. — Режим доступа: <http://e.lanbook.com/book/75386> — Загл. с экрана.

в) программное обеспечение и Интернет-ресурсы

- Microsoft Visual Studio 2010-2015

8. Материально-техническое обеспечение дисциплины

Имеются в наличии учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, промежуточной аттестации, а также помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет». Наличие рекомендованной литературы.

Программа составлена в соответствии с требованиями ФГОС ВО с учетом рекомендаций и ОПОП ВО по направлению 02.03.01 Математика и компьютерные науки.

Автор (ы) _____Штанюк А.А._____

Рецензент (ы) _____

Заведующий кафедрой_____Гергель В.П._____

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