МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Радиофизический факультет
(факультет / институт / филиал)

УТВЕРЖДАЮ:

Декан _____ Матросов В.В.

«____» _____ 2017 г.

Рабочая программа дисциплины

Б1.Б.15 Основы программирования
(наименование дисциплины (модуля))

Уровень высшего образования
бакалавриат
(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность
02.03.02 Фундаментальная информатика и информационные технологии
(указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы
Информационные системы и технологии
(указывается профиль / магистерская программа / специализация)

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

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

Нижний Новгород

2017 г.

Место и цели дисциплины в структуре ОПОП
Дисциплина «Основы программирования» относится к дисциплинам базовой части основной профессиональной образовательной программы по направлению 02.03.02 «Фундаментальная информатика и информационные технологии», преподается в 1 и 2 семестрах.

Целями освоения дисциплины являются:
Целью преподавания дисциплины «Основы программирования» является ознакомление студентов с основными понятиями информатики, методами программирования, простейшими алгоритмами, дающими представление о предмете в целом. Основной упор при этом делается на изучение методики постановки и решения вычислительных задач на современных ЭВМ, на формирование у студентов логически обоснованного подхода к выбору средств достижения результата и проведение анализа этого результата.

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

Формируемые компетенции (Код компетенции, этап формирования	**Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций**
ОПК-2. Способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий. (Этап формирования: начальный)	З1 (ОПК-2). Знать основы алгоритмизации и принципы использования алгоритмических языков программирования для постановки и решения прикладных задач на ЭВМ. У1 (ОПК-2). Уметь применять математические методы описания и исследования телекоммуникационных систем. В1 (ОПК-2). Владеть навыками использования инструкций командной строки для работы с файловой системой.
ОПК-3. Способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям. (Этап формирования: начальный)	З1 (ОПК-3). Знать принципы трансляции и верификации программ. З2 (ОПК-3). Знать особенности различных парадигм программирования. У1 (ОПК-3). Уметь использовать инструментальные средства программирования. У2 (ОПК-3). Уметь применять приобретенные знания в решении практических задач на примере традиционного набора простейших задач вычислительной математики. В1 (ОПК-3). Владеть базовыми средствами и навыками модульного программирования на языках С / С++.

2. Структура и содержание дисциплины «Основы программирования»

Объем дисциплины составляет 6 зачетных единиц, всего 216 часов, из которых 99 часов составляет контактная работа обучающегося с преподавателем (32 часа занятия лекционного типа, 64 часа занятия лабораторного типа, в том числе 4 часа – мероприятия текущего контроля успеваемости, 3 часа – мероприятия промежуточной аттестации), 117 часов составляет самостоятельная работа обучающегося.

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				**Самостоятельная работа обучающегося, часы**
		Занятия лекционного типа	**Занятия семинарского типа**	**Занятия лабораторного типа**	**Всего**	
1.1. Введение в дисциплину «Основы программирования».	2	1			1	1
1.2. Основы работы в ОС Windows и ОС Linux.	4	1		2	3	1
1.3. Введение в язык Си. Структура программы на языке Си. Примеры простых программ.	4	1		2	3	1
1.4. Основные этапы трансляции программы на языке Си. Задачи, решаемые препроцессором, компилятором, компоновщиком.	2	1			1	1
1.5. Типы данных в языке Си: базовые, производные и составные типы данных.	2	1			1	1
1.6. Константы и переменные в языке Си.	2	1			1	1
1.7. Представление данных в ЭВМ. Двоичное представление целочисленных данных.	3	1		1	2	1
1.8. Двоичное представление вещественных данных. Двоичное представление символьной информации.	3	1		1	2	1
1.9. Операции в языке Си.	7	1		4	5	2
1.10. Операторы языка Си.	7	1		4	5	2
1.11. Функции в языке Си.	8	1		4	5	3
1.12. Массивы в языке Си.	7	1		4	5	2
1.13. Указатели в языке Си. Работа со строками.	7	1		4	5	2
1.14. Структуры в языке Си. Средство описания типов typedef. Объединения в языке Си.	4	1		2	3	1
1.15. Средства динамического распределения памяти.	4	1		2	3	1
1.16. Обзор библиотечных функций языка Си.	5	1		2	3	2

2.1. Основы создания программ на языке C++. Компиляция. Директива include. Комментарии. Простейший ввод и вывод. Объявление переменных. Объявление простых функций и их использование.	9	1		2	3	6
2.2. Типы данных в C++, преобразование типов, арифметические операции.	9	1		2	3	6
2.3. Массивы. Строки в стиле C. Строки класса string. Указатели. Управление динамической памятью с помощью new и delete.	9	1		2	3	6
2.4. Структуры в C++.	9	1		2	3	6
2.5. Циклы for, while, do while.	9	1		2	3	6
2.6. Ссылочные переменные. Передача функции аргументов по ссылке. Аргументы по умолчанию.	18	2		4	6	12
2.7. Перегрузка функций. Шаблоны функций. Спецификации шаблонов функций.	18	2		4	6	12
2.8. Раздельная компиляция программ. Продолжительность хранения, область видимости и компоновка.	9	1		2	3	6
2.9. Определение и реализация класса. Открытый и закрытый доступ к классу. Данные-члены класса. Методы класса. Создание и использование объектов класса. Конструкторы и деструкторы класса. Функции-члены const. Указатель this. Создание массивов объектов.	18	2		4	6	12
2.10. Перегрузка операций. Дружественные функции. Перегрузка операции для вывода.	18	2		4	6	12
2.11. Динамическое выделение памяти для членов класса. Явные и неявные конструкторы копирования. Явные и неявные перегруженные операции присваивания. Статические члены класса. Указатели на объекты.	16	2		4	6	10
В т.ч. текущий контроль	4			4	4	
Промежуточная аттестация: **зачет, экзамен**						

3. Образовательные технологии

Образовательные технологии, способствующие формированию компетенций.

***используемые на занятиях лекционного типа*:**

- лекции с изложением учебного материала.

***используемые на занятиях лабораторного типа*:**

- решение типовых задач программирования с использованием технологии коллективной мыслительной деятельности.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает следующие виды:

- изучение дополнительных разделов дисциплины с использованием учебной литературы;
- изучение и тестирование библиотек языков программирования.

Текущий контроль усвоения материала проводится путем проведения опроса.

5. Фонд оценочных средств для промежуточной аттестации по дисциплине, включающий:

6.1. Перечень компетенций выпускников образовательной программы с указанием результатов обучения (знаний, умений, владений), характеризующих этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования.

Индикаторы компетенции	**Критерии оценивания (зачёт)**	
	«незачтено»	**«зачтено»**
Знания	Наличие грубых ошибок в основном материале	Знание основного материалом, возможно с рядом погрешностей
Умения	Наличие грубых ошибок при выполнении стандартных заданий	Способность выполнения всех стандартных заданий, возможно с незначительными погрешностями
Навыки	Отсутствие навыка	Достаточное владение навыком

Индикаторы компетенци	**Критерии оценивания (экзамен)**						
	«плохо»	**«неудовлетворительно»**	**«удовлетворительно»**	**«хорошо»**	**«очень хорошо»**	**«отлично»**	**«превосходно»**
Знания	Отсутствие знаний материала	Наличие грубых ошибок в основном материале	Знание основного материала с рядом негрубых ошибок	Знание основного материалом с рядом заметных погрешностей	Знание основного материала с незначительными погрешностями	Знание основного материала без ошибок и погрешностей	Знание основного и дополнительного материала без ошибок и погрешностей
Умения	Отсутствует способность решения стандартных задач	Наличие грубых ошибок при решении стандартных задач	Способность решения основных стандартных задач с существенными 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	Полное отсутствие навыка	Отсутствие навыка	Владение навыком в минимальном объёме	Посредственное владение навыком	Достаточное владение навыком	Хорошее владение навыком	Всестороннее владение навыком

6.2. Описание шкал оценивания.

Контроль качества усвоения студентами содержания дисциплины проводится в виде зачета и экзамена.

Зачет проводится в устной форме и заключаются в ответе студентом после предварительной подготовки на теоретические вопросы курса и решением практической задачи с последующим его обоснованием. По окончании ответа на вопросы в рамках тематики курса проводится собеседование в форме вопросов, на которые студент должен дать краткий ответ.

Итоговый контроль качества усвоения студентами содержания дисциплины проводится в виде экзамена, на котором определяется:

- уровень усвоения студентами основного учебного материала по дисциплине;
- уровень понимания студентами изученного материала;
- способности студентов использовать полученные знания для решения конкретных задач.

Критерии оценок (зачёт, экзамен).

Оценка	Уровень подготовки
Зачтено	В целом хорошая подготовка с возможными ошибками или недочетами. Студент дает полный ответ на все теоретические вопросы. Допускаются ошибки при ответах на дополнительные и уточняющие вопросы.
Не зачтено	Подготовка недостаточная и требует дополнительного изучения материала. Студент дает ошибочные ответы, как на теоретические вопросы билета, так и на дополнительные вопросы.

Оценка	Уровень подготовки
Превосходно	Высокий уровень подготовки, безупречное владение теоретическим материалом, студент демонстрирует творческий поход к решению нестандартных ситуаций. Студент дал полный и развернутый ответ на все теоретические вопросы билета, подтверждая теоретический материал практическими примерами.
Отлично	Высокий уровень подготовки с незначительными ошибками. Студент дал полный и развернутый ответ на все теоретические вопросы билета, подтверждает теоретический материал практическими примерами.
Очень хорошо	Хорошая подготовка. Студент дает ответ на все теоретические вопросы билета при наличии неточностей.
Хорошо	В целом хорошая подготовка с заметными ошибками или недочетами. Студент дает полный ответ на все теоретические вопросы билета при наличии неточностей. Допускаются ошибки при ответах на дополнительные и уточняющие вопросы экзаменатора.
Удовлетворительно	Минимально достаточный уровень подготовки. Студент показывает минимальный уровень теоретических знаний, делает существенные ошибки, но при ответах на наводящие вопросы, может правильно сориентироваться и в общих чертах дать правильный ответ.
Неудовлетворительно	Подготовка недостаточная и требует дополнительного изучения материала. Студент дает ошибочные ответы, как на теоретические вопросы билета, так и на дополнительные вопросы экзаменатора.
Плохо	Подготовка абсолютно недостаточная. Студент не отвечает на поставленные вопросы.

6.3. Критерии и процедуры оценивания результатов обучения по дисциплине, характеризующих этапы формирования компетенций.

Для оценивания результатов обучения в виде **знаний** используются следующие процедуры и технологии: экзамен, проводимый в письменной форме с дальнейшим индивидуальным собеседованием.

Для оценивания результатов обучения в виде **умений** и **навыков** используются результаты применения типовых алгоритмических решений в контексте их использования в реальных задачах.

6.4. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения, характеризующих этапы формирования компетенций и (или) для итогового контроля сформированности компетенции.

Типовые задания для текущего контроля успеваемости.

6.4.1. Задания для оценки компетенции «ОПК-3»:

1) Используя функции математической библиотеки, реализовать программу на языке Си для вычисления значений для $y(x, a)$ в соответствии со своим вариантом. Программа во время своей работы должна принимать значения переменных x и a, определять их принадлежность области допустимых значений. При возможности подсчета y как действительного числа – вывести его значение, в противном случае – вывести сообщение о причине невозможности нахождения значения переменной y (по вариантам).
2) Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран символов текста, располагающихся в заданном файле на нечётных позициях.
3) Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран чётных строк текста, содержащегося в заданном файле.
4) Программе задаются: имя входного текстового файла, начальный символ строк. Программа должна осуществлять вывод на экран содержащихся в заданном файле строк текста, начинающихся с указанного символа.
5) Программе задаются: целочисленные первый член и шаг арифметической прогрессии, количество элементов последовательности, имя выходного текстового файла. Программа должна осуществлять подсчёт и запись в текстовый файл значений элементов арифметической прогрессии.
6) Программе задаётся четырёхбайтовое целое число в шестнадцатеричном представлении. Программа должна осуществлять перемену местами старшего и младшего байтов заданного числа (например, 0xAABBCCDD→0xDDBBCCAA) и выводить на экран результат в шестнадцатеричном виде.
7) В командной строке программе задаются: первый целочисленный аргумент арифметической операции, символ, определяющий арифметическую операцию ('+', '–', '*', '/'), второй целочисленный аргумент арифметической операции. Программа должна осуществлять заданную арифметическую операцию с заданными целыми числами и выводить на экран результат. При невозможности выполнения операции деления на нуль на экран должно выводиться соответствующее сообщение об ошибке.
8) Программе задаётся буквенный символ из латинского алфавита. Программа должна осуществлять вывод на экран всех переменных окружения (и их значений), название которых начинается с заданной буквы.
9) Программе задаются N значений элементов целочисленного массива (величина N считается константой, определённой на этапе компиляции программы, например, при помощи директивы препроцессора #define N 10). Программа должна осуществлять поиск максимального числового значения среди всех N заданных целых чисел (элементов массива) с выводом на экран индекса и значения найденного максимального элемента массива.
10) Построить прототипы функций по заданным описаниям:
    а) tofu(...) принимает аргумент int и возвращает float;
    б) plot(...) принимает указатель на структуру tap как аргумент и возвращает строку;
    в) func2(...) принимает структуру boss в качестве аргумента и не возвращает ничего.
11) Предположим, что somearr – массив из 8 элементов double. Объявите указатель на первый элемент массива somearr и используйте этот указатель для отображения первого и последнего элементов массива.
12) Чему будет равно *ptr_arr после выполнения данного участка программы?

```
int arr[4] = {-2,-1,0,1};
int *ptr_arr = &arr[2];
ptr_arr++;
```

13) Вы хотите записать строчное значение в char carr[100] с клавиатуры. Вы знаете, что значение может включать пробельные символы. Какая из операций ввода, приведенных ниже, Вам подойдет:
    а) cin << carr;
    б) cin.getline(carr,100);
    в) cin >> carr;

Типовые задания (оценочные средства), выносимые на зачет.

6.4.2. Задания для оценки компетенции «ОПК-2»:

1) Команды для работы с файловой системой в командной строке Unix: ls, cd. Примеры использования.
2) Команды для работы с файловой системой в командной строке Unix: cp, mv, rm. Примеры использования.
3) Команды для работы с файлами в командной строке Unix: cat, more, less. Примеры использования.
4) Команды для работы с файловой системой в командной строке Windows: dir, cd. Примеры использования.
5) Команды для работы с файловой системой в командной строке Windows: copy, del, move. Примеры использования.
6) Команды для работы с файловой системой в командной строке Windows: mkdir, move. Примеры использования.

6.4.3. Задания для оценки компетенции «ОПК-3»:

1) Целочисленные простые типы данных в языке программирования Си (char, int). Форматы двоичного представления и диапазоны значений.
2) Вещественнозначные простые типы данных в языке программирования Си (float, double). Форматы двоичного представления и диапазоны значений.
3) Использование квалификаторов при объявлении переменных.
4) Объявление переменных в языке программирования Си. Способы инициализации объявленных переменных.
5) Время жизни и область видимости переменных в языке программирования Си. Статические переменные. Локальные и глобальные переменные.
6) Константы в языке программирования Си, способы задания констант. Целочисленные (десятичные, восьмеричные, шестнадцатеричные) и действительные константы. Именованные константы (директива #define).
7) Арифметические операции в языке программирования Си. Операции инкремента и декремента.
8) Операции отношения и логические операции в языке программирования Си.
9) Битовые операции в языке программирования Си.
10) Операторы ветвления в языке программирования Си. Безусловный (goto) и условный (if … else) переходы, блок-схемное представление.
11) Операторы циклов с предусловием (while) и с постусловием (do … while) в языке программирования Си. Блок-схемное представление циклических конструкций.
12) Оператор цикла for в языке программирования Си. Эквивалентное описание циклической конструкции при помощи оператора while.

13) Функции в языке программирования Си. Декларирование, описание и вызов. Особенности передачи параметров функций.
14) Массивы в языке программирования Си. Объявление и использование одномерных массивов.
15) Представление строк в языке программирования Си как массивов символов.
16) Многомерные массивы в языке программирования Си. Объявление и использование двумерных массивов.
17) Указатели в языке программирования Си. Операции с указателями, адресная арифметика. Взаимосвязь массивов и указателей.
18) Основные функции, используемые при работе с текстовыми файлами (открытие, закрытие, чтение, запись и пр.) в библиотеке стандартного ввода / вывода языка программирования Си.

Типовые задания (оценочные средства), выносимые на экзамен.

6.4.4. Задания для оценки компетенции «ОПК-2»:

1) Напишите функцию, возвращающую значение типа void, которая присваивает всем элементам массива типа int заданное число типа int. Функция должна принимать три аргумента: указатель на первый элемент в массиве, размер массива и это число.
2) Напишите функцию с именем mystrlen, которая подсчитывает количество символов в строке. Строка не должна изменяться. Не используйте стандартные функции strlen, size, length.
3) Напишите функцию, принимающую имя массива типа double и его размер в качестве аргументов и возвращающую наибольшее и наименьшее значение, которое содержится в этом массиве. Функция не должна изменять содержимое массива.
4) Напишите функцию, которая меняет порядок элементов массива типа double на обратный (первый элемент становится последним, последний – первым, второй – предпоследним, предпоследний – вторым и т.д.) Функция должна принимать два аргумента: указатель на первый элемент в массиве и размер массива. Покажите пример вызова написанной функции.

6.4.5. Задания для оценки компетенции «ОПК-3»:

1) Имеется шаблон вида:
   template <typename T>
   T sqr(T x){return x*x;}
   Определите тип генерируемой функции и возвращаемое значение при следующих вызовах: sqr(3), sqr(5UL), sqr(2e2).
2) Что будет выведено на экран?
   int hex_val = 0x10;
   std::cout << hex_val << " " << std::hex << hex_val << std::endl;
3) Что напечатает код:
   cout << (int *) "Hello, world!" << endl;.
   а) номер символа 'H' в ASCII коде
   б) Hello, world!
   в) H
   г) адрес, соответствующий символу 'H'
4) Дана программа:
   #include <iostream>
   using namespace std;

```
class XY {
  private:
    double x, y;
  public:
    XY(){}
    ~XY(){}
    XY(double x1, double y1) {x = x1; y = y1;}
    void show() const {cout  << "x=" << x  << ", y=" << y << endl;}
    XY operator* (double a) const { return XY(a*x, a*y); }
    XY operator~() const { return XY(-x, y); }
    friend XY operator* (double a, const XY &v) { return XY(a*v.x, a*v.y); }
};
int main(void) {
  XY v1(2.0, 1.0);
  XY v2;
  v2 = 2.0 * v1;
  XY v3 = ~v1;
  v3.show();
  return 0;
}
```

а) Укажите функцию, которая вызывается строчкой v2 = 2.0 * v1; внутри main.

б) Что будет выведено на экран в результате выполнения программы?

5) Что напечатает приведенная ниже программа? (Подсказка: стандартная функция strlen возвращает длину строки без учета нулевого символа.)

```
#include <iostream>
#include <cstring>
using namespace std;
template <typename T> T myfunc(T x, T y) {
  T p = (x > y) ? x : y;
  return p;
}
template <> const char * myfunc(const char *x, const char *y) {
  const char * v = (strlen(x) > strlen(y)) ? x : y;
  return v;
}
int main(void) {
  const char *u="abc";
  const char *v="defg";
  double x=5.5, y=6;
  int i=4, j=3;
  int ij = myfunc(i, j);
  double xy = myfunc(x, y);
  const char *uv = myfunc(u, v);
  cout << "ij=" << ij << endl;
  cout << "xy=" << xy << endl;
  cout << "uv=" << uv << endl;
}
```

6.5. Методические материалы, определяющие процедуры оценивания.

- Положение «О проведении текущего контроля успеваемости и промежуточной аттестации обучающихся в ННГУ», утверждённое приказом ректора ННГУ от 13.02.2014 г. №55-ОД.
- Положение «О фонде оценочных средств», утвержденное приказом ректора ННГУ от 10.06.2015 г. №247-ОД.

6. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Трофимов В.В. Информатика. Том 1: 3-е изд., пер. и доп. – М.: Издательство Юрайт, 2017. – 553 с. [Электронный ресурс: https://biblio-online.ru/book/F0FE998E-C747-4ABB-84E3-07A146765A50]
2. Трофимов В.В. Информатика. Том 2: 3-е изд., пер. и доп. – М.: Издательство Юрайт, 2017. – 406 с. [Электронный ресурс: https://biblio-online.ru/book/5A795D83-C63B-4210-93C5-B3AC5093CC91]
3. Кудинов Ю.И., Пащенко Ф.Ф. Основы современной информатики: учебное пособие. – СПб.: Издательство «Лань», 2017. – 256 с. [Электронный ресурс: https://e.lanbook.com/reader/book/86016/]
4. Степанов А.Н. Информатика: базовый курс. – СПб.: Питер, 2011. – 719 с.
5. Соболь Б.В. и др. Информатика: учебник. – Ростов-на-Дону: Феникс, 2006. – 448 с.
6. Керниган Б., Ритчи Д. Язык программирования С: пер. с англ. – М.: Вильямс, 2006. – 304 с.
7. Прата С. Язык программирования С: лекции и упражнения: пер. с англ. – Киев: Диа-Софт, 2000. – 432 с.
8. Огнева М.В., Кудрина Е.В. Программирование на языке С++: практический курс. Учебное пособие для бакалавриата и специалитета. – М.: Издательство Юрайт, 2017. – 335 с. [Электронный ресурс: https://biblio-online.ru/book/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4]
9. Зыков С.В. Программирование. – М.: Издательство Юрайт, 2017. – 320 с. [Электронный ресурс: https://biblio-online.ru/book/122D27F3-13E4-4095-8946-C619F0FCC5C3]

б) дополнительная литература:

1. Трофимов В.В., Павловская Т.А. Алгоритмизация и программирование. – М.: Издательство Юрайт, 2017. – 137 с. [Электронный ресурс: https://biblio-online.ru/book/B08DB966-3F96-4B5A-B030-E3CD9085CED4]
2. Колдаев В.Д. Основы алгоритмизации и программирования: учебное пособие / под ред. проф. Л.Г. Гагариной. – М.: ИД «ФОРУМ» : ИНФРА-М, 2017. – 416 с. [Электронный ресурс: http://znanium.com/bookread2.php?book=902236]
3. Акулов О.А., Медведев Н.В. Информатика: базовый курс: учебник для вузов. – М.: Омега-Л, 2008. – 574 с.
4. Клеменс Б. Язык С в XXI веке. – М.: ДМК Пресс, 2015. – 376 с. [Электронный ресурс: http://www.studentlibrary.ru/book/ISBN9785970601013.html]

в) программное обеспечение и Интернет-ресурсы:

1. Интегрированная среда разработки программного обеспечения Microsoft Visual Studio.
2. Средства компиляции GNU Compiler Collection в составе ОС Linux.
3. Среда разработки программного обеспечения Geany в составе ОС Linux.

7. Материально-техническое обеспечение дисциплины

Аудиторный фонд ННГУ для проведения лекций.
Компьютерные класс лаборатории «ЭВМ».

Программа составлена в соответствии с требованиями ФГОС ВО с учетом рекомендаций и ОПОП ВПО по направлению 02.03.02 «Фундаментальная информатика и информационные технологии».

Автор (ы) _____А.А. Горбунов

_____ А.А. Рябов

_____ А.В. Маслов

Рецензент_____ С.Н. Жуков

Заведующий кафедрой «Безопасность
информационных систем» _____Л.Ю. Ротков

Программа одобрена на заседании методической комиссии Радиофизического факультета. Протокол № 04/17 от «30» августа 2017 года.