


Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»

Институт непрерывного образования российских и иностранных граждан

УТВЕРЖДАЮ
Директор ИНО



Е.Л. Сырцова

10 октября 2018 г.

№ 04-04-2018-0158-0363

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Основы программирования»**

**по дополнительной профессиональной программе-
программе профессиональной переподготовки
«Системное программирование и компьютерные технологии»**

Рабочую программу разработал: преподаватель кафедры ПМИ ВятГУ
Прозорова Татьяна Георгиевна

© Вятский государственный университет, 2018

© Прозорова Т.Г., 2018

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Актуальность дисциплины

Дисциплина «Основы программирования» является одной из важнейших в подготовке слушателей в области информационных технологий. Изучаемый материал является основой для изучения всех последующих дисциплин, связанных с разработкой программных продуктов.

Дисциплина формирует у слушателей знания, умения и навыки в области изучения основных понятий языков программирования; синтаксиса, семантики, формальных способов описания языков программирования; типов данных, способов и механизмов управления данными; современных методов и парадигм программирования.

Цель – изучение основных понятий языков программирования; синтаксиса, семантики, формальных способов описания языков программирования; типов данных, способов и механизмов управления данными; современных методов и парадигм программирования. Знания, полученные в результате изучения данной дисциплины необходимы для создания фундамента освоения новых языков программирования.

Задачи:

- подготовить специалистов, владеющих современными методами и средствами программирования;
- выработать у слушателей навыки использования языков программирования для создания систем обработки данных, обоснованного выбора средств программирования;
- приобрести практические навыки решения задач от построения алгоритма ее решения, до выполнения ее на компьютере.

Компетенции слушателя, формируемые в результате освоения учебной дисциплины / модуля

В результате освоения учебной дисциплины обучающийся должен демонстрировать следующие результаты обучения:

Виды деятельности	Профессиональные компетенции	Практический опыт	Умения	Знания
ВД-1	К 1 – способность разрабатывать компоненты системных программных продуктов	- разработка драйверов устройств; - разработка системных утилит; - создание инструментальных средств	- применять языки программирования, языки целевой аппаратной платформы, языки программирования	- средства программирования и их классификацию; - архитектуру сред программирования; - классификацию средств

		программирования.	я на разработку системных утилит, определенных в техническом задании; - создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов; - осуществлять отладку драйверов устройств, программных продуктов, утилит операционной системы; - применять языки программирования высокого уровня, определенные в техническом задании.	программирования; - архитектуру целевой аппаратной платформы; - методы и основные этапы трансляции; - технологии разработки драйверов.
--	--	-------------------	--	---

1.2 Содержание учебной дисциплины

Объем учебной дисциплины и виды учебной работы

Форма обучения	Общий объем (трудоемкость) Часов	в том числе аудиторная контактная работа обучающихся с преподавателем, час					Самостоятельная работа, час	Форма промежуточной аттестации
		Всего	Лекции	Практические (семинарские) занятия	Лабораторные занятия	Консультации		
очная	60	48	12	36	-	-	12	экзамен

Тематический план

№ п/п	Основные разделы и темы учебной дисциплины	Часы		Самостоятельная работа
		Лекции	Практические занятия	
1.	Базовые средства языка C++	8	28	8
2.	Модульное программирование	4	8	4
	Итого:	12	36	12

Матрица соотнесения разделов / тем учебной дисциплины / модуля и формируемых в них компетенций

Разделы/темы учебной дисциплины	Компетенции		
	Количество часов	К-1	Общее количество компетенций
Базовые средства языка C++	44	+	1
Модульное программирование	16	+	1
Итого	60		

Краткое содержание учебной дисциплины

Тема 1. Базовые средства языка C++

История возникновения языка Си, его характеристики, состав языка (алфавит языка, идентификаторы, ключевые слова, знаки операций, константы, комментарии).

Понятие типов данных (концепция типа данных, основные типы данных, структура программы), переменных и выражений (переменные, операции, выражения).

Базовые конструкции структурного программирования (операторы ветвления if и switch, операторы цикла for, while, do...while, операторы передачи управления break, continue, return).

Понятие указателей и массивов: создание статических и динамических массивов. Алгоритмы обработки массивов: линейный и бинарный поиск, сортировка элементов массива, перестановка элементов массива.

Пользовательские типы данных «структура» и «перечисление». Использование структур и перечислений при разработке программ.

Тема 2. Модульное программирование

Интерфейс функции: передача параметров в функции, передача массивов в функции, возвращаемые значения. Функции с переменным числом параметров, с параметрами по умолчанию, без параметров.

Рекурсивные функции: достоинства и недостатки, примеры реализации с рекурсией и без. Функция main().

Создание многомодульной программы: этапы, директивы препроцессора, пространства имен.

Работа с файлами заданной структуры.

2. Учебно-методическое обеспечение дисциплины

2.1. Методические рекомендации для преподавателя

Выбор методов и средств обучения, образовательных технологий осуществляется преподавателем исходя из необходимости достижения обучающимися планируемых результатов освоения дисциплины, а также с учетом индивидуальных возможностей обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья.

Изучение дисциплины следует начинать с проработки настоящей рабочей программы, методических указаний и разработок, указанных в программе, особое внимание уделить целям, задачам, структуре и содержанию дисциплины.

2.2. Методические указания для слушателей

Успешное освоение учебной дисциплины предполагает активное, творческое участие слушателя на всех этапах ее освоения путем планомерной, повседневной работы. Слушатель обязан посещать лекции и практические занятия, получать консультации преподавателя и выполнять самостоятельную работу.

Главной задачей каждой лекции является раскрытие сущности темы и анализ ее основных положений. Содержание лекций определяется настоящей рабочей программой дисциплины.

Предполагается, что слушатели приходят на лекции, предварительно проработав соответствующий учебный материал по источникам, рекомендуемым программой.

Целью практических занятий является проверка уровня понимания слушателями вопросов, рассмотренных на лекциях и в учебной литературе, степени и качества усвоения материала; применение теоретических знаний в реальной практике решения задач; восполнение пробелов в пройденной теоретической части курса и оказания помощи в его освоении.

Практические занятия в равной мере направлены на совершенствование индивидуальных навыков решения теоретических и прикладных задач, выработку навыков интеллектуальной работы, а также ведения дискуссий. Конкретные пропорции разных видов работы в группе, а также способы их оценки определяются преподавателем, ведущим занятия.

На практических занятиях под руководством преподавателя слушатели обсуждают дискуссионные вопросы, отвечают на вопросы тестов, закрепляя приобретенные знания, выполняют практические задания и т.п. Для успешного проведения практического занятия слушателям следует тщательно подготовиться.

Основной формой подготовки слушателей к практическим занятиям является самостоятельная работа с учебно-методическими материалами, научной литературой, статистическими данными и т.п.

Самостоятельная работа слушателей включает в себя выполнение различного рода заданий (изучение учебной и научной литературы,

материалов лекций, систематизацию прочитанного материала, подготовку контрольной работы, решение задач и т.п.), которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По каждой теме учебной дисциплины преподаватель предлагает слушателям перечень заданий для самостоятельной работы. Самостоятельная работа по учебной дисциплине может осуществляться в различных формах (например, подготовка докладов; написание рефератов; публикация тезисов; научных статей; подготовка и защита проекта; другие).

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны выполняться самостоятельно либо группой и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению.

Регулярно рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Результатом самостоятельной работы должно стать формирование у слушателей определенных знаний, умений, навыков, компетенций.

При проведении промежуточной аттестации слушателя учитываются результаты текущей аттестации в течение периода обучения.

Процедура оценивания результатов освоения учебной дисциплины (модуля) осуществляется на основе действующего Положения об организации текущего контроля успеваемости и промежуточной аттестации обучающихся ВятГУ.

Для приобретения требуемых компетенций, хороших знаний и высокой оценки по дисциплине слушателям необходимо выполнять все виды работ своевременно в течение всего периода обучения.

Самостоятельная работа

Самостоятельная работа слушателей включает изучение материалов лекций, учебников, проработку тем, вынесенных на самостоятельное изучение, подготовку к экзамену.

Слушатель изучает материал лекций по конспекту, в котором изложены основные понятия по теме. С помощью законодательных документов и учебников слушатель прорабатывает и углубляет знания по теме лекции.

3. Учебно-методическое обеспечение учебной дисциплины

Литература

1. Кирнос, В. Н. Информатика II. Основы алгоритмизации и программирования на языке C++ [Электронный ресурс] : учебно-методическое пособие / В.Н. Кирнос. - Томск : Эль Контент, 2013. – 160 с. Полный текст находится в ЭБС "Университетская библиотека ONLINE".
2. Язык Си и особенности работы с ним [Электронный ресурс] : учебное пособие / Н.И. Костюкова. - Москва : Интернет-Университет Информационных Технологий, 2006. - 207 с.. - (Основы информационных технологий)
3. Программирование на языке Си [Электронный ресурс] : учебное пособие / Р.Ю. Царев. - Красноярск : Сибирский федеральный университет, 2014. - 108 с.
4. Программирование на языке C++ и среде Qt Creator [Электронный ресурс] : [учебник] / Е. Р. Алексеев [и др.]. - Москва : ALT Linux, 2015. - 448 с.. - (Библиотека ALT Linux)
- 5.

Перечень специализированных аудиторий (лабораторий)

Вид занятий	Назначение аудитории
Лекции, практика	Учебная аудитория, оснащенная мультимедийным оборудованием и персональными компьютерами
Самостоятельная работа	Читальные залы библиотеки

Перечень специализированного оборудования

Перечень используемого оборудования
Мультимедиа проектор
Ноутбук
Экран с электроприводом
Персональные компьютеры

Перечень информационных технологий, используемых при осуществлении образовательного процесса по учебной дисциплине

№ п/п	Наименование ПО	Краткая характеристика назначения ПО
1	Office Professional Plus 2013 Russian OLP NL Academic	Пакет приложений для работы с различными типами документов: текстами, электронными таблицами, базами данных, презентациями
2	Windows 7 Professional and Professional K	Операционная система
3	Microsoft Visual Studio 2015 Professional	Среда разработки программного обеспечения, включающая в себя текстовый редактор

		исходного кода и компиляторы языков программирования
4	Kaspersky Endpoint Security для бизнеса	Антивирусное программное обеспечение

4. Материалы, устанавливающие содержание и порядок проведения промежуточных аттестаций

Форма контроля по дисциплине – экзамен по билетам (в классической форме). Билет содержит два теоретических вопроса и одну практическую задачу.

К сдаче экзамена допускаются все слушатели, проходящие обучение на данной ДПП, вне зависимости от результатов текущего контроля успеваемости и посещаемости занятий, при этом, результаты текущего контроля успеваемости могут быть использованы преподавателем при оценке уровня усвоения обучающимися знаний, приобретения умений, навыков и сформированности компетенций в результате изучения учебной дисциплины.

На подготовку к ответу отводится не более 45 минут. Время, отводимое на ответ по билету, не должно превышать 20 минут, включая ответы на дополнительные вопросы.

Если обучающийся отказался от ответа на выбранный билет, то преподаватель может предложить ему другой билет, с выставлением пониженной на один балл оценки.

Во время экзамена обучающиеся могут пользоваться рабочей программой учебного курса, а с разрешения экзаменатора – справочниками, картами, таблицами и другими пособиями.

Вопросы для подготовки к экзамену

1. Алгоритм. Свойства алгоритмов. Способы записи алгоритмов. Структурная схема алгоритма.
2. Перечислить символы, используемые при составлении структурных схем алгоритмов. Примеры.
3. Перечислить базовые конструкции структурного программирования.
4. Составить схему алгоритма нахождения корней квадратного уравнения.
5. Дать определение понятию "лексема", "ключевые слова". Описать правила составления идентификаторов.
6. Перечислить и охарактеризовать основные типы данных языка С.
7. Переменная, именованная константа. Их описание на языке С.
8. Как определить область действия идентификатора?
9. Перечислить основные классы памяти и охарактеризовать их.
10. Перечислить основные арифметические операции, используемые при кодировании программ на языке С.
11. Перечислить логические операции, используемые при кодировании программ на языке С. Где и зачем используются логические выражения.

12. Каким образом в языке C производится вывод информации в консоль. Перечислить операции вывода языка C.
13. Каким образом в языке C производится ввод информации с консоли. Перечислить операции ввода языка C.
14. Каким образом в языке C можно вывести на консоль текст на русском языке?
15. Описать синтаксис оператора ветвления if, оператора множественного ветвления switch.
16. Что такое итерация?
17. Чем отличаются циклы с предусловием и с постусловием?
18. В чем особенность цикла со счетчиком for? Описать синтаксис оператора for. Оператор break. Оператор continue. Оператор return.
19. Массив. Как описать массив в программе на языке C. Многомерный массив.
20. Что такое указатель? Зачем нужны указатели в языке C?
21. Перечислить способы выделения памяти в языке C.
22. Что такое функция в языке C. Интерфейс функции. Тип функции.
23. Дать определение понятию "инкапсуляция".
24. Зачем нужны встроенные функции? Как их описать в программе? Выделить достоинства и недостатки встроенных функций.
25. Перечислить и охарактеризуйте способы передачи параметров в функцию. В каких случаях после выполнения функции значение переменной-параметра может измениться?
26. Как избежать непредусмотренного изменения параметра в теле функции?
27. Что такое рекурсивная функция? Выделить достоинства и недостатки рекурсивных функций.
28. Зачем нужны параметры функции main? Привести пример использования параметров функции main.
29. Зачем нужны пользовательские типы данных в языке C/C++?
30. Перечислить этапы создания исполняемого файла.
31. Что такое препроцессор языка C/C++?
32. Что такое директивы препроцессора? Что позволяет сделать директива include? директива define? директивы условной компиляции?
33. Что такое компоновщик? Что такое компилятор языка программирования? Чем компиляторы отличаются от интерпретаторов?

Практические задания для подготовки к экзамену

1. Сформировать двумерный массив размера 6×8. Заполнить массив таким образом, чтобы каждый элемент был равен сумме индексов ячейки, в которой он находится. Вывести элементы массива на экран.
2. Заполнить массив из задания 1 случайными целыми числами. Найти сумму элементов на главной диагонали. Вывести на экран.
3. Сформировать одномерный массив, элементы которого равны максимальным отрицательным элементам строк.

4. Вывести на экран таблицу умножения чисел от 1 до 10.

	1	2	...	10
1	1	2	...	10
2	2	4	...	20
...
10	10	20	...	100

5. По номеру дня недели (0 – вс, 1 – пн, ... 6 – сб) вывести его название. В случае некорректного ввода вывести сообщение.
6. Создать одномерный массив из 10 вещественных элементов. Заполнить случайно. Отсортировать по возрастанию модулей элементов.
7. Проверить, пройдет ли кирпич размерами $h \times w \times l$ в круглое отверстие радиуса r .
8. Заполнить массив из 15 элементов случайными буквами английского алфавита. Вывести полученную строку на экран. Развернуть строку. Вывести результат.
9. Вывести на экран прямоугольный треугольник высоты n из звёздочек
- Пример. $n = 5$*

```
      *
     **
    ***
   ****
  *****
```

10. Написать функцию, генерирующую целое число из диапазона $[-20, 20]$.
11. Написать функцию, генерирующую вещественное число из диапазона $[-5, 5]$ с тремя знаками после запятой.
12. Написать фрагмент кода, меняющий местами две переменные с использованием третьей переменной.
13. Написать фрагмент кода, меняющий местами две переменные без использования вспомогательной переменной.
14. Написать фрагмент кода, переставляющий указанный элемент в конец массива. Порядок остальных элементов при этом измениться не должен.
15. Написать фрагмент кода, сортирующий одномерный массив по возрастанию значений элементов.
16. Написать фрагмент кода, в котором описывается одномерный динамический массив.
17. Написать фрагмент кода, в котором происходит обращение к k -ому элементу массива двумя способами.
18. Выделить память под одномерный динамический массив из n элементов типа `unsigned char`. Сколько байт будет выделено?
19. Написать фрагмент кода, в котором освобождается память из-под одномерного динамического массива.
20. Написать фрагмент кода, в котором описывается двумерный динамический массив.
21. Выделить память под двумерный динамический массив размера $n \times m$ элементов типа `float`. Сколько байт будет выделено?

22. Написать фрагмент кода, в котором освобождается память из-под двумерного динамического массива.
23. Написать программу вычисления определителя матрицы (рекурсия).
24. Написать программу нахождения заданного числа в отсортированном массиве.
25. Написать программу сортировки массива.