

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
Высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)



А.А.Панфилов

« 17 » 04

2015г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»

Направление подготовки 01.03.02. Прикладная математика и информатика

Профиль/программа подготовки

Уровень высшего образования бакалавриат

Форма обучения очная

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточно- го контроля (экз./зачет)
3	6/216	36	-	36	99	Экзамен (45 часов)
Итого	6/216	36	-	36	99	Экзамен (45 часов)

Владимир 2015

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целями освоения дисциплины (модуля) «Языки и методы программирования» являются изучение студентами парадигм программирования, современных языков и методов программирования. Выработка у студентов умения самостоятельно реализовывать алгоритмы на языке программирования высокого уровня для решения прикладных задач, используя при этом наиболее подходящие методы программирования, уметь классифицировать данные и выбирать нужный класс данных для решения определенной задачи.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина входит в базовую часть дисциплин ОПОП. Содержание дисциплины охватывает круг вопросов, связанных с разработкой и реализацией алгоритмов решения прикладных задач программирования, создания программных продуктов. Для успешного освоения курса студенты должны владеть знаниями и навыками, полученными ими при изучении школьного курса «Информатика и ИКТ», необходимо знание основных алгоритмических структур, представлений о машинной арифметике. Знания и умения, приобретенные студентами в результате изучения дисциплины, будут ими использоваться при изучении следующих курсов: «Объектно-ориентированное программирование», «Математическое моделирование», «Интеллектуальный анализ данных», «Разработка кросс-платформенных приложений», написании выпускной квалификационной работы.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанные с прикладной математикой и информатикой (ОПК-1);

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 (ОПК-3);

способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения (ПК-7);

В результате освоения дисциплины студент должен демонстрировать освоение указанных компетенций по дескрипторам «знания, умения, владения», в соответствии с тематическими модулями дисциплины, применять полученные знания в последующем обучении и профессиональной деятельности:

1) Знать: основные парадигмы программирования и сферы их применения (ОПК-1); основные алгоритмические структуры (ОПК-3); основы современных языков программирования (ПК-7).

2) Уметь: применять основные парадигмы программирования для решения задач прикладной математикой, опираясь на базовые знания естественных наук, математики и информатики (ОПК-1); самостоятельно разрабатывать алгоритмические и программные решения (ОПК-3); разрабатывать алгоритмические решения и реализовывать их на языке программирования высокого уровня (ПК-7).

3) Владеть: умением использовать парадигмы программирования для решения задач прикладной математики (ОПК-1); навыками разработки алгоритмических и программных решений задач прикладной математики (ОПК-3); навыками разработки и применения алгоритмических и программных решений в области системного и прикладного программного обеспечения (ПК-7).

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

№ п/п	Раздел (тема) дисциплины	Семестр		Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации (по семестрам)
				Неделя семестра	Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС		
1.	История развития языков программирования, обзор современных языков и методов программирования. Структура программы на языке C++. Базовые типы данных. Операторы, приоритет операций.	3	1-5	10	-	6	-	20	-	5/25	Рейтинг-контроль-1
2.	Типы данных определяемые пользователем	3	4-11	12	-	14	-	20	-	6/25	
3.	Функции	3	11-13	4	-	4	-	15	-	2/25	Рейтинг-контроль-2
4.	Динамические структуры данных	3	14-15	4	-	4	-	15	-	4/50	
5.	Файлы	3	16	2	-	2	-	10	-	1/25	
6.	Введение в язык Python.	3	16-18	4	-	6	-	19	-	2/25	Рейтинг-контроль-3
Всего:		3	18	36	-	36	-	99	-	20/27	экзамен

Лекции.

1. Введение. История развития языков программирования, обзор современных языков и методов программирования. (2 часа)
2. Структура программы на языке C++ (2 часа)
3. Классификация типов данных. Базовые типы данных. (2 часа)
4. Переменные и выражения. Операторы, приоритет операций. (2 часа)
5. Область действия и классы памяти. (2 часа)
6. Указатели. Ссылки. (4 часа)
7. Массивы. (4 часа)
8. Пользовательские типы данных. Перечисления. Структуры. Объединения. (4 часа)
9. Функции. Объявление и определение. Передача параметров. Возвращаемое значение. (2 часа)
10. Функции. Перегрузка функций. Шаблоны функций. (2 часа)

11. Динамические структуры данных. Линейный списки. Стек Очередь. Бинарные деревья (4 часа)
12. Работа с файлами. (2 часа)
13. Введение в язык Python. (4 часа)

Лабораторные работы.

1. Операторы. Приоритет операций (6 часов)
2. Указатели и ссылки. (4 часа)
3. Массивы данных. (4 часа)
4. Составные типы данных. Структуры. (6 часов)
5. Функции. (4 часа)
6. Динамические структуры данных. (4 часа)
7. Файлы.(2 часа)
8. Основы языка Python (6 часов)

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Для успешного освоения студентами данного курса необходимо использование активных и интерактивных методов обучения с уделением должного внимания имитационным моделям, позволяющих наиболее эффективно организовать процесс учебно-познавательной и исследовательской деятельности студентов, способствующему наиболее глубокому овладению компетенциями.

Сочетание традиционных образовательных технологий в форме лекции с интерактивными занятиями проведении мероприятий контроля текущих знаний студентов (задания рейтинг контроля, своевременная защита лабораторных работ), позволяют наиболее глубоко вовлечь студентов в процесс познавательной, научно-исследовательской деятельности, эффективно и равномерно распределить время самостоятельной работы студентов в течение семестра.

Во время занятий предусмотрено проведение диалога и дискуссий на лекциях по различным вопросам работы со сложными структурами данных и развития современных языков программирования, ряду других вопросов.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Рейтинг-контроль №1

Вариант 1.

1. `int a=5; int b=a++; int c=++a;`
Каковы значения переменных a,b,c после выполнения этой строчки кода?
2. `int a=7; int b=27;`
`int k=(a>b)?printf("a+b=%i",a+b): printf("a-b=%i",a-b);`
Чему равно k?
3. `signed char ch=129;`
`int a=ch++;`
Каковы значения переменных a и ch?
4. `unsigned char i = 0, j = 256;`
`int k = j || i; int g = ~j;`
Каковы значения k и g?

Вариант 2.

1. `int a=9; int b=a--; int c=--a;`
Чему равны значения переменных a,b,c после выполнения этой строчки кода?

2. `int a=37; int b=27;`
`int k=(a>b)?printf("a+b=%i",a+b++): printf("a-b=%i",a-b);`
 Чему равно k?
3. `unsigned char ch=255;`
`int a=ch++;`
 Чему равны значения переменных a и ch?
4. `signed char i =0, j = 374;`
`int k = j || i;`
`int g = ~j;`
 Чему равны значения переменных k и g?

Рейтинг-контроль №2

Вариант 1.

1. Написать фрагмент кода и схематически отобразить результат его работы. Объявить указатель на `int`, переменную типа `int`, настроить указатель на эту переменную и записать в нее через указатель число 10.
2. Объявить константный указатель на `int`, проинициализировать, сделать поясняющий рисунок.
3. Объявить одномерный массив `int` из пяти элементов, проинициализировать при объявлении. Сохранить в целочисленную переменную значение первого элемента массива. Показать два способа (индексное выражение и используя имя массива, как указатель)
4. Опишите синтаксис, работу и пример использования оператора цикла `while`

Вариант 2.

1. Написать фрагмент кода и схематически отобразить результат его работы. Объявить указатель на `float`, переменную типа `float`, настроить указатель на эту переменную и записать в нее через указатель число 10,5.
2. Объявить константный указатель на указатель на `int`, проинициализировать, сделать поясняющий рисунок.
3. Объявить одномерный массив `int` из пяти элементов, проинициализировать при объявлении. Сохранить в целочисленную переменную значение пятого элемента массива. Показать два способа (индексное выражение и используя имя массива, как указатель)
4. Опишите синтаксис, работу и пример использования оператора цикла `do while`

Рейтинг-контроль №3

Вариант 1

Организовать линейный список в виде стека с возможностью добавления и удаления элемента, вставки по ключу.

Вариант 2

Организовать линейный список в виде очереди с возможностью добавления и удаления элемента вставки по ключу.

Вопросы к экзамену.

1. Понятие тип данных. Базовые типы языка C++.
2. Условный оператор. Синтаксис, пример использования.
3. Оператор выбора. Синтаксис, пример использования.
4. Цикл с предусловием. Синтаксис, пример использования.
5. Цикл с постусловием. Синтаксис, пример использования.
6. Цикл с параметром. Синтаксис, пример использования.
7. Массив. Определение, объявление, способы обращения к элементам.

8. Структуры. Определение, описание, объявление, способы обращения к полям.
9. Функции. Объявление, определение, вызов, формальные, фактические параметры.
10. Перегрузка функций.
11. Шаблоны функций.
12. Типы данных языка Python.
13. Локальные и глобальные переменные.
14. Организация циклов на Python.
15. Основные операции и операторы Python
16. Списки Python.
17. Массивы Python.
18. Печать данных по формату.
19. Работа с файлами на Python.
20. Функции в языке программирования Python. Параметры и аргументы функций.

Задания для самостоятельной работы студентов.

1. Найти периметр и площадь квадрата, факториал числа, интеграл (способы передачи параметров: по значению, по указателю, по ссылке, передача функции в качестве параметра).
2. Умножение матриц. Передача массивов в качестве параметров. 2 способа: статические и динамические массивы. Представление двумерного массива в виде вектора с использованием арифметики указателей.
3. Телефонный справочник. Массив структур. Реализовать функции: добавить, удалить, найти, вывести список. Использовать передачу параметров (массив и количество элементов, например).
4. Телефонный справочник. Связные списки. Реализовать функции: добавить элемент, добавить в указанное место, удалить заданный, найти, вывести список, обратить список. Использовать передачу параметров.
5. Найти периметр и площадь квадрата, факториал числа, интеграл (способы передачи параметров: глобальные переменные, по значению, по указателю, по ссылке, передача функции в качестве параметра). Объяснить в каждом случае, как передаются параметры, какой способ лучше для вычисления факториала.
6. Умножение матриц. Передача массивов в качестве параметров. 2 способа: статические и динамические массивы. Представление двумерного массива в виде вектора с использованием арифметики указателей.
7. Телефонный справочник. Использовать массив структур. Реализовать функции: добавить, удалить, найти, вывести список. Использовать передачу параметров (массив и количество элементов, например).
8. Телефонный справочник. Организовать с использованием связанного списка. Реализовать функции: добавить элемент, добавить в указанное место, удалить заданный, найти, вывести список, обратить список.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

а) основная литература:

1. Златопольский Д.М. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс] / Д. М. Златопольский. - 2-е изд. (эл.). - 223 с. 2012 www.studentlibrary.ru
2. С. М. Окулов Программирование в алгоритмах изд. (эл.). - М. : БИНОМ. Лаборатория знаний, -383 с. 2014 www.studentlibrary.ru
3. Давыдова Н.А. Программирование [Электронный ресурс]: учебное пособие/ Давыдова Н.А., Боровская Е.В.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2015.— 239 с.— Режим доступа: <http://www.iprbookshop.ru/6485>.— ЭБС «IPRbooks», 2015

б) дополнительная литература:

1. В.В. Подбельский Язык С#. Базовый курс: учеб. Пособие. М.: Финансы и статистика, - 408 с. 2015 www.studentlibrary.ru
2. В.В. Подбельский. Язык Си#. Решение задач: учеб. Пособие. - М.: Финансы и статистика. 296 с. 2014 <http://www.studentlibrary.ru>
3. Магда Ю.С. Программирование и отладка С/С++ приложений для микроконтроллеров ARM Издательство ДМК-пресс. 2012 <http://www.studentlibrary.ru/>

в) периодические издания:

1. «Информационные технологии» Ежемесячный теоретический и прикладной научно-технический журнал ISSN 1684-6400
2. Журнал «Вестник компьютерных и информационных технологий» ISSN 1810-7206
3. Журнал «Вестник ВлГУ» ISSN 2307-3241.


в) интернет-ресурсы:

1. Официальный сайт <https://www.python.org/>
2. Электронное пособие в открытом доступе. Столяров А.В. Введение в язык С++. МАКС-ПРЕСС 2011. <http://www.stolyarov.info>
3. Электронный ресурс, содержащий пошаговую инструкцию по установке и самоучитель Python <https://pythonworld.ru/osnovy/skachat-python.html>
4. Онлайн справочник программиста на С и С++ <http://www.c-cpp.ru/books>

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для успешного освоения дисциплины необходима аудитория для проведения лекционных занятий, имеющая необходимое количество посадочных мест и оснащенная доской для мела или маркера, компьютерный класс для проведения лабораторных работ, имеющих необходимое количество рабочих мест, с набором базового программного обеспечения для разработки программ на языке программирования С++.

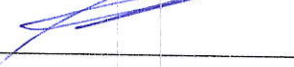
Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 01.03.02. Прикладная математика и информатика

Рабочую программу составил старший преподаватель каф. ФиПМ Шишкина М.В. 

Рецензент
(представитель работодателя) ген. директор ООО "РС Сервис" Квасов Д.С.
(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПМ

Протокол № 11А от 17.04.15 года

Заведующий кафедрой ФиПМ  С.М. Аракелян
(ФИО, подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления _____

Протокол № 11А от 17.04.15 года

Председатель комиссии _____ С.М. Аракелян
(ФИО, подпись)

ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ (МОДУЛЯ)

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____