

УТВЕРЖДАЮ
Директор ИК
_____ С.А.Байдали
«__» _____ 2016 г.

**БАЗОВАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»**

Направление ООП – 01.03.02 «Прикладная математика и информатика»

Профиль подготовки – Компьютерное моделирование

Применение математических методов к решению инженерных и экономических задач

Квалификация (степень) – бакалавр

Базовый учебный план приема – 2016 г.

Курс – 1, семестр – 2

Количество кредитов – 4

Код дисциплины – ДИСЦ.В.М6

Виды учебной деятельности	Временной ресурс по очной форме обучения
Лекции, ч	48
Лабораторные занятия, ч	48
Практические занятия, ч	
Аудиторные занятия, ч	96
Самостоятельная работа, ч	48
ИТОГО, ч	144

Вид промежуточной аттестации – зачет, диф. зачет (КР)

Обеспечивающее подразделение – кафедра прикладной математики

Заведующий кафедрой _____ О.М. Гергет

Руководитель ООП _____ Г.Е. Шевелев

Преподаватель _____ С.А. Рыбалка

1. Цели освоения модуля (дисциплины)

В результате освоения данной дисциплины бакалавр приобретает знания, умения и навыки, обеспечивающие достижение целей Ц1, Ц2 и Ц5 основной образовательной программы «Прикладная математика и информатика».

Основные цели освоения дисциплины:

1. Развитие компьютерной грамотности в условиях роста темпов информатизации общества и приобретение профессиональных навыков в области разработки и решения задач с использованием современных компьютерных технологий;
2. Освоение современных методов представления и обработки данных;
3. Приобретение практических навыков в конструировании классов;
4. Освоение теоретических основ построения языков программирования.

2. Место модуля (дисциплины) в структуре ООП

Дисциплина входит в междисциплинарный профессиональный модуль вариативной части (ДИСЦ.В.М). Она является постреквизитом для дисциплин междисциплинарного профессионального модуля вариативной части (ДИСЦ.В.М) – Практикум на ЭВМ (первый семестр), модуля естественнонаучных и математических дисциплин базовой части (ДИСЦ.Б.М) – Информатика (первый семестр).

Является пререквизитом для дисциплин – (ДИСЦ.В.М5) «Практикум на ЭВМ» (четвертый семестр) и (ДИСЦ.В.М11) «Операционные системы» (четвертый семестр).

3. Результаты освоения модуля (дисциплины)

При изучении дисциплины бакалавры должны познакомиться с основами разработки современных приложений; научиться создавать собственные программные продукты на современном уровне; приобрести базовые навыки использования механизмов абстракции при разработке программ; знать структуру компиляторов, схему компиляции, назначения отдельных узлов схемы компиляции; приобрести теоретические сведения о построении формальных языков и грамматик.

После изучения данной дисциплины бакалавры приобретают знания, умения и опыт, соответствующие результатам основной образовательной программы: Р1, Р3 и Р9¹. Соответствие результатам освоения дисциплины «Языки и методы программирования» формируемым компетенциям представлено в таблице.

¹ Расшифровка кодов результатов обучения представлена в Основной образовательной программе подготовки бакалавров по направлению (специальности) 010400, Прикладная математика и информатика

Формируемые компетенции в соответствии с ООП	Результаты освоения дисциплины
<p>ПК1 способностью демонстрации общенаучных базовых знаний естественных наук, математики и информатики, понимание основных фактов, концепций, принципов теорий, связанных с прикладной математикой и информатикой</p> <p>ПК10 способностью применять в профессиональной деятельности современные языки программирования и языки баз данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии</p>	<p>3.1.1 Основы и концепции разработки приложений с использованием элементов объектно-ориентированного, визуального, модульного, событийного программирования.</p> <p>3.1.2 Принципы построения компиляторов, схему компиляции, назначения отдельных узлов схемы компиляции.</p> <p>3.1.3 Основ и концепций формализации методов класса.</p>
	<p>В.3.1 Опыт создания приложений с использованием простых АТД (классов) и управлением классом через элементы интерфейса.</p> <p>В.1.1. Опыт применять в профессиональной деятельности парадигмы современного программирования.</p> <p>В.1.2. Опыт теоретического анализа грамматик языка с целью классификации.</p>
<p>ПК12 способностью составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы</p>	<p>У.1.1. Уметь составлять план работ и контролировать выполняемую работу.</p> <p>У.1.2 Уметь планировать необходимые для выполнения работы информационные ресурсы.</p> <p>В.9.1. Опыт оценивать результаты собственной работы.</p>
<p>ОК14 способностью использовать в научной и познавательной деятельности, а также в социальной сфере профессиональные навыки работы с информационными и компьютерными технологиями</p>	<p>У.1.3. Владеть приемами создания приложений в среде С++ Builder.</p> <p>У.1.4. Уметь связывать формальное определение языка программирования с технологией, лежащей в основе методов трансляции.</p>

4. Структура и содержание модуля (дисциплины)

4.1. Структура модуля (дисциплины) по разделам, формам организации и контроля обучения

№	Название раздела/темы	Аудиторная работа (час)		СРС (час)	Колл, Контр.Р	Итого
		Лекции	Лаб. зан.			
1.	Основы разработки приложений с использованием современных технологий программирования в среде С++ Builder	6	16	18	Отчеты по лаб. работам. Коллоквиум	40
2	Механизмы абстракции в С++	8	4	8	Отчеты по лаб. работам. Рубежный контроль	20
3.	Элементы модульного программирования в С++	4	4	4	Отчеты по лаб. работам.	12
4	Реализация ввода-вывода при помощи потоков. Построение структур данных	6	10	4	Отчеты по лаб. работам.	20
5	Основные вычислительные алгоритмы	8	14	10	Отчеты по лаб. работам. Устный опрос.	32
6	Теоретические основы построения компиляторов	16		4	Зач	20
Ит		48	48	48		144

ого						
-----	--	--	--	--	--	--

При сдаче отчетов и защите лабораторных работ проводится устный опрос.

4.2. Содержание разделов дисциплины

Раздел 1. Основы разработки приложений с использованием современных технологий программирования в среде C++ Builder

Общая характеристика проекта C++ Builder. Структура приложения. Компиляция и запуск.

Основы визуального проектирования. Понятие компонента. Компоненты для ввода данных и отображения результатов. Общие свойства компонентов.

Управление проектом. Элементы событийно – ориентированного программирования. Использование технологий современного программирования в C++ Builder.

Лабораторная работа 1. Создание приложения, имитирующего работу простейшего калькулятора в среде C++ Builder.

Лабораторная работа 2. Вычисление по формулам в среде C++ Builder.

Лабораторная работа 3. Отображение текущей даты и фиксация времени решения задачи на форме приложения.

Лабораторная работа 4. Создание многооконных приложений в среде C++ Builder.

Лабораторная работа 5. Использование компонентов: Мемо, ListBox, ComboBox.

Раздел 2. Механизмы абстракции в C++

Процедуры и функции как механизмы абстракции. Абстракции через спецификацию. Абстракции через параметризацию. Механизмы параметризации (ссылки и значения). Параметры типов и параметризованные типы. Параметры «по умолчанию». Использование спецификации const при передаче параметров. Функции «in-line». Указатели на функцию.

Абстрактный тип данных (АТД). Определение. Концепция объектно-ориентированное программирования — от физического объекта к программной модели.

Схема процесса создания программ для решения прикладных задач. Особенности разработки АТД в C++. Указатели «this». Дружественные функции. Конструктор копирования. Перегрузка операторов. Дружественные классы.

Лабораторная работа 6. Использование компонентов StringGrid для отображения матриц.

Лабораторная работа 7. Управление классом через элементы интерфейса.

Раздел 3. Элементы модульного программирования в C++

Определение модуля. Директивы препроцессора для подключения файлов. Условная компиляция. Разделение интерфейса и реализации. Структура хэдер – файла. Структура реализационного файла. Технология подключения модулей в среде C++ Builder.

Лабораторная работа 8. Использование элементов модульного программирования при разработке классов.

Раздел 4. Реализация ввода-вывода при помощи потоков. Построение динамических структур данных. (6 часов)

Понятие потока. Общая характеристика иерархии классов, отвечающих за ввод-вывод данных в C++. Спецификации конструкторов классов ifstream, ofstream, fstream.

Понятие динамических структур данных. Деки, стеки, очереди, списки. Хеш-таблицы.

Лабораторная работа 9. Использование файлов в работе со структурами данных.

Лабораторная работа 10. Сохранение структурных данных в текстовых файлах.

Лабораторная работа 11. Настройка свойств компонента StringGrid при отображении в нем структурных данных.

Раздел 5. Основные вычислительные алгоритмы

Основные классические алгоритмы сортировки массивов в оперативной памяти: метод наименьшего элемента, метод плавающего пузырька, метод прямого включения, Шейкер-сортировка, сортировка Шелла, сортировка с вычисляемыми адресами.

Итеративные и рекурсивные алгоритмы. Понятие эвристических алгоритмов. Алгоритмы поиска и технология программирования с отходом назад. Быстрая сортировка.

Лабораторная работа 12. Создание меню и диалогов.

Лабораторная работа 13. Использование «блокнота» для отображения информации в среде C++ Builder.

Раздел 6. Теоретические основы построения компиляторов

Основные функции трансляторов. Типы трансляторов. Классификация трансляторов. Концептуальная схема работы компилятора. Понятие прохода.

Формальные языки и грамматики. Основные определения. Способы записи правил грамматики. Классификация языков и грамматик с использованием иерархии Ноама Хомского. Цепочки вывода.

Распознаватели. Классификация распознавателей.

Регулярные языки, конечные автоматы. Приведение регулярных грамматик к автоматному виду. Пример построения конечного автомата.

Синтаксический анализ. КС – языки и магазинные автоматы. Нисходящий синтаксический анализ. Восходящий синтаксический анализ.

4.3. Распределение компетенций по разделам дисциплины

Распределение по разделам дисциплины планируемых результатов обучения по базовой образовательной программе, формируемых в рамках данной дисциплины:

№	Формируемые компетенции	Разделы дисциплины					
		1	2	3	4	5	6
1	З.1.1	x		x	x		
2	З.1.2						x
3	З.1.3					x	
4	У.1.1		x	x	x	x	
5	У.1.2			x	x		
6	У.3.1	x		x	x		
7	У.1.3	x		x	x		
8	У.1.4						x
9	В.1.1	x	x	x			
10	В.1.2						x
11	В.3.1	x	x	x	x	x	
12	В.9.1				x	x	

5. Образовательные технологии

При освоении дисциплины используются следующие сочетания видов учебной работы с методами и формами активизации познавательной деятельности бакалавров для достижения запланированных результатов обучения и формирования компетенций.

Методы и формы активизации деятельности	Виды учебной деятельности			
	ЛК	ЛБ	КР	СРС
Дискуссия	x			
IT-методы	x	x	x	
Работа в команде				
Опережающая самостоятельная работа		x	x	
Индивидуальное обучение		x	x	x
Проблемное обучение		x	x	
Поисковый метод			x	x
Разбор кейсов		x		
Обучение на основе опыта	x	x	x	

Для достижения поставленных целей преподавания дисциплины реализуются следующие средства, способы и организационные мероприятия:

- изучение теоретического материала на лекциях;
- самостоятельное изучение теоретического материала дисциплины с использованием *Internet* ресурсов, информационных баз, методических разработок, специальной учебной литературы;
- закрепление теоретического материала при проведении лабораторных работ;
- закрепление теоретического материала в курсовой работе;
- подготовка к коллоквиуму, рубежному и итоговому контролю.

6. Организация и учебно-методическое обеспечение самостоятельной работы студентов (СРС)

6.1. Текущая и опережающая СРС

Текущая и опережающая самостоятельная работа студентов направлена на углубление и закрепление знаний, а также практических умений. К ней относятся:

- изучение тем, вынесенных на самостоятельную проработку;
- работа бакалавров с лекционным материалом;
- поиск и анализ литературы и электронных источников информации по проблеме, поставленной в задаче для курсовой работы;
- изучение методических указаний к лабораторным работам и к курсовой работе;
- выполнение курсовой работы;
- подготовка к коллоквиумам, к зачету.

6.1.1. Перечень тем, выносимых на самостоятельную проработку²:

- «Технологические элементы разработки приложений в среде C++ Builder» [6, 8];
- «Характеристики и свойства языков программирования»;
- «Алгоритмические стратегии» [10].

6.2. Творческая проблемно-ориентированная самостоятельная работа

Творческая проблемно-ориентированная самостоятельная работа ориентирована на развитие интеллектуальных умений, комплекса универсальных (общекультурных) и профессиональных компетенций, повышение творческого потенциала студентов и заключается:

- в поиске, анализе и структурировании информации, необходимой для выполнения курсовой работы;
- в составлении плана работ и контролировании выполняемой работы;
- в выполнении курсовой работы;
- в участие в олимпиадах по программированию;
- в анализ теоретического материала по теме, определенной преподавателем;
- в разработке и создании АТД, составление спецификаций для методов класса, формализация методов класса;
- в проведение расчетов с использованием АТД;
- в подготовке презентации для защиты курсовой работы.

6.2.1. Примеры темы курсовой работы

1. Разработка класса «Множество» на основе бинарных данных.
2. Разработка класса, позволяющего отображать картографическую информацию о системах дорог, связывающих между собой географические объекты двух видов: населенные пункты (города и поселки) и архитектурные сооружения (здания/дома).

² Теоретические сведения хранятся в виде файлов на локальном сервере кафедры ПМ

3. Известно, что физическое, эмоциональное и умственное состояние человека изменяется со дня его рождения циклически с периодом 23, 28 и 33 дня, соответственно. Написать программу построения графиков биоритмов на заданный интервал времени.
4. Разработка АТД «Полином», в котором требуется, в том числе, включить следующие операции: сложение полиномов, умножение полинома на число, получить производную от полинома. Для хранения полиномов использовать стеки.
5. Разработка класса «Домашняя фонотека» (названия кассет, компакт – дисков, авторы и исполнители песен). Предусмотреть ввод и вывод информации по запросу (например: список произведений данного исполнителя, самый популярный автор,....)³.

6.2.2. Требования к курсовой работе:

- 1) В основу приложения должна быть положена иерархия классов (АТД), разработанных пользователем.
- 2) Реализация АТД должна быть оформлена в виде модулей и включена в C++ Builder.
- 3) Формализацию основных вычислительных методов классов должно оформить в виде отдельных блок – схем с последующим включением их в пояснительную записку отдельным параграфом.
- 4) Отчет должен быть подготовлен средствами текстового процессора Microsoft Word в соответствии с требованиями ГОСТа и требования ТПУ
- 5) В отчете в разделе приложений должны быть представлены страницы с кодами программных модулей.
- 6) Программный код должен сопровождаться грамотными комментариями, позволяющими оценивать работу алгоритма.

6.3. Контроль самостоятельной работы

Оценка результатов самостоятельной работы организуется как единство двух форм: самоконтроль и контроль со стороны преподавателей. Оценка успеваемости бакалавров осуществляется по результатам:

- самостоятельного (под контролем преподавателя) выполнения лабораторных работ;
- самостоятельного (под контролем преподавателя) выполнения курсовых работ;
- устного опроса при защите отчетов по лабораторной работе;
- устного опроса по темам СРС, проводимого в форме коллоквиума;
- устного опроса при защите курсовой работы.

6.4. Учебно-методическое обеспечение самостоятельной работы студентов

Образовательные ресурсы, рекомендуемые для использования при самостоятельной работе студентов:

- Рыбалка С.А., Шкатова Г.И. C++ Builder. Задачи и решения. Учебное пособие. – Томск: Изд-во ТПУ, 2010. – 486 с.
- С.А.Рыбалка, Г.И.Шкатова. Методические указания «Языки программирования и методы трансляции» — Томск: изд. ТПУ, 2000 г. — 88 с.
- С.А.Рыбалка, Г.И.Шкатова. Методические указания к выполнению лабораторных работ, 2008 г. (в электронном виде).

³ Можно взять самые популярные группы, скачать информацию (в том числе фото) о них, например если рок, то охарактеризовать, что это за стиль.

- http://portal.tpu.ru/SHARED/g/GISH/Teacher_Work/LPr1.
- <http://www.cyberguru.ru> – информационный сайт для разработчиков программного обеспечения на С++ и других системах программирования.

7. Средства (ФОС) текущей и итоговой оценки качества освоения модуля (дисциплины)

Оценка качества освоения дисциплины, а также оценка успеваемости бакалавров осуществляется по результатам:

- самостоятельного (под контролем преподавателя) выполнения лабораторной работы;
- ответов на вопросы рубежного, итогового контролей;
- анализа подготовленных магистрантами рефератов,
- устного опроса при защите курсовых работ;
- устного опроса при защите отчетов по лабораторным работам и во время «зачета».

7.1. Банк данных теоретических вопросов

Примерный перечень вопросов для оценки текущей успеваемости и промежуточной аттестации студентов по итогам освоения модуля (дисциплины):

1. Классификация языков программирования?
2. Характеристика и свойства языков программирования?
3. Основные парадигмы современного программирования?
4. Как понимать термин «процедурная абстракция»?
5. Механизмы процедурной абстракции?
6. Что содержится в спецификации процедуры?
7. Механизмы параметризации в С++,
8. Понятие абстрактного типа данных?
9. Как использовать поля структуры в вычислениях?
10. Как изменятся параметры у функции на С++, если она станет полем структуры?
11. Из каких соображений выбираются поля класса на С++?
12. Назначение конструктора класса?
13. Назначение деструктора класса?
14. Назначение конструктора копирования?
15. Как определяются типы полей класса, и из каких соображений задаются их имена?
16. Как определяется список параметров методов класса?
17. Что представляет собой поток ввода – вывода?
18. Почему нужно проверять возможность ошибки при открытии файла?
19. Как определяют, какие методы инкапсулированы или встроены в класс?
20. С какими классами связаны операции ввода – вывода?
21. Определение списка? Стека? Очереди? Дека?
22. Основные операции со списками?
23. Определение узла списка?
24. Когда и где отводится память под переменную ссылочного типа?
25. В каких операциях проявляются преимущества при работе со списками? С двусвязными списками?
26. Что означает понятие «переопределить операцию»? Когда это имеет смысл?
27. Сколько параметров может иметь operator – функция на С++, если она является членом класса?

28. Сколько параметров может иметь дружественная оператор – функция?
29. Как реализован механизм модульного программирования в C++?
30. Основные технологии современного программирования?
31. Механизм событийного программирования?
32. Понятие проекта?
33. Структура проекта в C++ Builder?
34. Что вкладывается в понятие «визуального программирования»?
35. Определение транслятора?
36. Виды трансляторов?
37. Условная трансляция в C++?
38. Структурная схема транслятора?
39. Что такое «лексема»? Лексический анализ?
40. Какую работу выполняет сканер?
41. Семантический анализ программы?
42. Цепочки, язык, грамматики, лексемы?
43. Цепочки вывода?
44. Определение распознавателя?
45. Классификация распознавателей?
46. Конечные автоматы?
47. Организация таблиц символов?
48. Классификация грамматик?

9. Учебно-методическое и информационное обеспечение модуля (дисциплины)

Основная литература

1. Рыбалка С.А., Шкатова Г.И. C++ Builder. Задачи и решения. Учебное пособие. – Томск: Изд-во ТПУ, 2010. – 486 с.
2. С.А.Рыбалка, Г.И.Шкатова. Методические указания «Языки программирования и методы трансляции» — Томск: изд. ТПУ, 2000 г. – 88 с.
3. С.А.Рыбалка, Г.И.Шкатова. Методические указания к выполнению лабораторных работ, 2008 г. (в электронном виде).
4. Компаниец Р.И., Маньков Е.В., Филатов Н.Е.. Системное программирование. Основы построения трансляторов. – СПб.:КОРОНА, 2004. – 256 с.

Дополнительная литература

5. Мозговой М.В. Классика программирования: АЛГОРИТМЫ, ЯЗЫКИ, АВТОМАТЫ, КОМПИЛЯТОРЫ. Практический подход. – СПб.: Наука и техника, 2006. – 320с.
6. Пахомов Б.И. C/C++ и Borland C++ Builder для начинающих. –Спб.: БХВ-Петербург, 2006, – 736 с.: ил.
7. Гордеев А.В., Молчанов А.Ю. Системное программное обеспечение. – СПб.: Питер, 2003, – 640 с.: ил.
8. Боровский А. C++ и Borland C++ Builder. Самоучитель. –Спб.: БХВ-Петербург, 2005, – 256 с.: ил.
9. Серебряков В.А., Галочкин М.П. Основы конструирования компиляторов. – М.: Эдиториал УРСС, 2001. – 224 с.

10. Альфред В. Ахо, Джон Хопкрофт, Джеффри Д. Ульман. Структуры данных и алгоритмы. – М. Вильямс, 2003. – 382 с.

Программное обеспечение

1. Операционная система Windows 2000/NT/XP.
2. Антивирусные программы. Архиваторы.
3. Microsoft Office 2003/2007.
4. Системы программирования Borland C++ и C++ Builder.

10. Материально-техническое обеспечение модуля (дисциплины)

Выполнение лабораторных работ происходит в компьютерных классах 102, 103, 104 здания ИК ТПУ.

Программа составлена на основе Стандарта ООП ТПУ в соответствии с требованиями ФГОС по направлению 01.03.02 Прикладная математика и информатика

Программа одобрена на заседании кафедры ПМ

(протокол № ____ от «__» _____ 2016 г.).

Автор: к.т.н., доцент Рыбалка С.А.