

Подписано электронной подписью:  
Вержицкий Данил Григорьевич  
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»  
Дата и время: 2024-02-21 00:00:00  
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Кемеровский государственный университет»  
Новокузнецкий институт (филиал)  
федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Кемеровский государственный университет»

Факультет физико-математический и технолого-экономический  
Профилирующая кафедра теории и методики преподавания информатики



### Рабочая программа дисциплины (модуля)

### **Б1.В.ДВ.08.01 Решение задач по программированию повышенной сложности**

*(Наименование дисциплины (модуля))*

Направление подготовки (специальность)

**44.03.01 Педагогическое образование**

Направленность (профиль) подготовки

**Информатика**

Программа: прикладной бакалавриат

Форма обучения

**Очная**

Год набора 2015

**Новокузнецк**

**2018**

## Лист внесения изменений

### Сведения об утверждении:

утверждена Ученым советом факультета

(протокол Ученого совета факультета № 6 от 3.03.2016 )

на 20\_16\_\_ год

Одобрена на заседании методической комиссии

протокол методической комиссии факультета № 6 от 18.02.2016 )

Одобрена на заседании обеспечивающей кафедры

протокол № 7 от 16.03.2016 ) Можаров М.С. (Ф. И.О. зав. кафедрой) / 

(подпись)

### Изменения по годам:

На 2017\_ год

утвержден (а) Ученым советом факультета

(протокол Ученого совета факультета № 7 от 16.03.2017 )

на 2017 год

Одобрен (а) на заседании методической комиссии

протокол методической комиссии факультета № 7 от 15.03.2017 )

Одобрен (а) на заседании обеспечивающей кафедры ТиМПИ

протокол № 8 от 02.03.2017 ) Можаров М.С. (Ф. И.О. зав. кафедрой) /  (подпись)

### Изменения по годам:

На 2018\_ год

утвержден (а) Ученым советом факультета

(протокол Ученого совета факультета № 6 от 15.02.2018 )

на 2018 год

Одобрен (а) на заседании методической комиссии

протокол методической комиссии факультета № 6 от 07.02.2018 )

Одобрен (а) на заседании обеспечивающей кафедры ТиМПИ

протокол № 5 от 19.01.2018 ) Можаров М.С. (Ф. И.О. зав. кафедрой) /  (подпись)

## СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения основной образовательной программы «Информатика» .....	4
2. Место дисциплины в структуре ОПОП бакалавриата .....	6
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся.....	6
3.1. Объем дисциплины (модуля) по видам учебных занятий (в часах) .....	6
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий.....	7
4.1. Разделы дисциплины (модуля) и трудоемкость по видам учебных занятий (в академических часах).....	7
4.2. Содержание дисциплины (модуля), структурированное по темам (разделам) .....	8
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю) .....	9
6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю) .....	11
6.1. Паспорт фонда оценочных средств по дисциплине (модулю).....	11
6.2. Типовые контрольные задания или иные материалы .....	11
6.2.1. Зачет .....	11
6.2.2. Наименование оценочного средства* (в соответствии с таблицей 6.1).....	13
6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций .....	16
7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля) .....	17
а) основная учебная литература:.....	17
б) дополнительная учебная литература:.....	17
8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля).....	17
9. Методические указания для обучающихся по освоению дисциплины (модуля) .....	17
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости) .....	18
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю) .....	18
12. Иные сведения и (или) материалы.....	18
12.1. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья.....	18
12.2. Занятия, проводимые в интерактивных формах .....	19

**1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения основной образовательной программы «Информатика»**

В результате освоения ОПОП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине (модулю):

<i>Коды компетенции</i>	<b>Результаты освоения ОПОП</b> <i>Содержание компетенций*</i>	<b>Перечень планируемых результатов обучения по дисциплине</b>
СПК-2	способен использовать математический аппарат, методологию программирования и современные компьютерные технологии для реализации аналитических и технологических решений в области программного обеспечения и компьютерной обработки информации	<p><b>Знать:</b>  методы и приемы формализации и алгоритмизации задач;  синтаксис языков программирования (Алгоритмический язык, Basic, Pascal, Python, C, Java, Prolog, Lisp), особенности программирования на выбранном языке, стандартные библиотеки языка программирования;  структуры данных и алгоритмы решения типовых задач, области и способы их применения;  методологии разработки программного обеспечения;  технологии программирования;  методы и приемы отладки программного кода, типы и форматы сообщений об ошибках, предупреждений;</p> <p><b>Уметь:</b>  использовать методы и приемы формализации и алгоритмизации поставленных задач;  использовать функциональные возможности компиляторов, трансляторов, отладчиков и интегрированных сред разработки для написания и отладки программного кода;  применять стандартные алгоритмы в соответствующих областях;  применять выбранные языки программирования для написания программного кода;  использовать выбранную среду программирования и средства системы управления базами данных;  выявлять ошибки в программном коде, применять методы и приемы отладки программного кода, интерпретировать сообщения об ошибках и предупреждения;</p> <p><b>Владеть:</b>  навыками составления формализованных описаний решений поставленных задач;  навыками разработки алгоритмов решения поставленных задач;  опытом применения выбранных языков</p>

		<p>программирования для написания программного кода;          владеть методами анализа, проверки и отладки исходного программного кода;          интерфейсом и функциональными возможностями Case-средств для структурного и объектно-ориентированного проектирования;          современными формализованными математическими, информационно-логическими и логико-семантическими моделями и методами представления, сбора и обработки информации;          методами программирования и навыками работы с математическими пакетами для решения практических задач хранения и обработки информации.</p>
ПК-4	<p>способностью использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемого предмета</p>	<p>Знать:          сущность личностных, метапредметных и предметных результатов обучения;          понятие «качество учебно-воспитательного процесса»;          основные характеристики и способы формирования безопасной развивающей образовательной среды;          специфику общего образования и особенности организации образовательного пространства в условиях образовательной организации; основные психолого-педагогические подходы к проектированию и организации образовательного пространства;          способы для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемого учебного предмета;          современные педагогические технологии реализации компетентного подхода с учетом возрастных и индивидуальных особенностей обучающихся; методы и технологии поликультурного, дифференцированного и развивающего обучения.</p> <p>Уметь:          применять современные образовательные технологии, включая информационные, а также цифровые образовательные ресурсы для достижения личностных, метапредметных и предметных результатов обучения;          разрабатывать и реализовывать проблемное обучение, осуществлять связь обучения по предмету (курсу, программе) с практикой, обсуждать с обучающимися актуальные события современности;          разрабатывать и реализовывать программы развития универсальных учебных действий, образцов и ценностей социального поведения;          поддерживать в детском коллективе деловую, дружелюбную атмосферу для обеспечения безопасной развивающей образовательной среды.</p> <p>Владеть:          навыками планирования и организации учебно-воспитательного процесса, ориентированного на достижение личностных, метапредметных и предметных результатов обучения;          навыками регулирования поведения обучающихся для обеспечения безопасной развивающей образовательной среды.</p>

## 2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Решение задач по программированию повышенной сложности» входит в вариативную часть профессионального цикла программы бакалавриата с кодом (Б.1.В) и изучается на 4 курсе.

Требования к входным знаниям и умениям: необходимо пройти обучение по дисциплинам «Математическая логика и теория алгоритмов», «Программирование».

Освоение данной дисциплины является необходимой основой для последующего изучения дисциплин вариативной части профессионального цикла.

Дисциплина (модуль) изучается на 4 курсе (ах) в 8 семестре (ах).

## 3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины (модуля) составляет 2 зачетных единиц (ЗЕТ), 72 академических часов.

### 3.1. Объем дисциплины (модуля) по видам учебных занятий (в часах)

Объём дисциплины	Всего часов	
	для очной формы обучения	для заочной формы обучения
Общая трудоемкость дисциплины	72	
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	30	
Аудиторная работа (всего**):	30	
в т. числе:		
Лекции	10	
Семинары, практические занятия		
Практикумы		
Лабораторные работы	20	
Занятия в интерактивной форме	16	
Внеаудиторная работа (всего**):	42	
В том числе, индивидуальная работа обучающихся с преподавателем:		
Курсовое проектирование		
Групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем		
Творческая работа (эссе)		
Самостоятельная работа обучающихся (всего**)	42	
Вид промежуточной аттестации обучающегося	Зачет	

<b>Объём дисциплины</b>	<b>Всего часов</b>	
	для очной формы обучения	для заочной формы обучения
(зачет / экзамен****)		

**4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий**

**4.1. Разделы дисциплины (модуля) и трудоемкость по видам учебных занятий (в академических часах)**

**для заочной (очно-заочной) формы обучения**

№ п/п	Раздел дисциплины	Общая трудоёмкость (часов)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах)			Формы текущего контроля успеваемости
			аудиторные учебные занятия		самостоятельная работа обучающихся	
		всего	лекции	семинары, практические занятия		
<b>1. Модульное программирование. Программирование абстрактных типов данных</b>						
1.	Процедуры и функции. Модули.	26		2	24	Лабораторная работа
2.	Организация динамических структур данных (абстрактных типов данных): списки, стеки, очереди, деревья.	32		2	30	Решение индивидуальных задач
<b>2. Объектно-ориентированное программирование</b>						
3.	Реализация абстракций данных методами объектно-ориентированного программирования.	36		4	32	Решение индивидуальных задач
4.	Объектно-событийное и объектно-ориентированное программирование.	46		6	40	Решение индивидуальных задач

**для очной формы обучения**

№ п/п	Раздел дисциплины	Общая трудоем- кость (часах)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах)			Формы текущего контроля успеваемости
			аудиторные учебные занятия		самостоятель- ная работа обучающихся	
		всего	лекции	семинары, практические занятия		
<b>1. Модульное программирование. Программирование абстрактных типов данных</b>						
5.	Процедуры и функции. Модули.	34		14	20	Лабораторная работа
6.	Организация динамических структур данных (абстрактных типов данных): списки, стеки, очереди, деревья.	36		16	20	Решение индивидуальных задач
<b>2. Объектно-ориентированное программирование</b>						
7.	Реализация абстракций данных методами объектно-ориентированного программирования.	36		16	20	Решение индивидуальных задач
8.	Объектно-событийное и объектно-ориентированное программирование.	38		18	20	Решение индивидуальных задач

#### **4.2 Содержание дисциплины (модуля), структурированное по темам (разделам)**

№ п/п	Наименование раздела дисциплины	Содержание
<i>Темы лабораторных занятий</i>		
<b>1. Модульное программирование. Программирование абстрактных типов данных</b>		
1	Процедуры, функции. Разработка и вызов подпрограмм в среде Gambas.	
2	Изучение внутренней структуры модулей, создание модуля и подключение его к проекту.	
3	Многооконные приложения. Создание модальных и немодальных окон в среде Gambas.	
4		
<b>2. Организация динамических структур данных (абстрактных типов данных): списки, стеки, очереди, деревья.</b>		
5	Простые списки.	

№ п/п	Наименование раздела дисциплины	Содержание
6		Неупорядоченные списки. Связанные списки.
7		Стеки.
8		Очереди.
9		Представления деревьев. Обход дерева.
10		Работа с древовидными иерархическими структурами данных.
11		Типы данных record и variant, работа с сеткой строк TStringGrid в среде Gambas.
3. Реализация абстракций данных методами объектно-ориентированного программирования.		
12		Абстрактные типы и структуры данных.
13		Классы, объекты, поля, методы.
14		Конструкторы и деструкторы.
15		Свойства и методы объектов.
16		Раннее связывание и позднее связывание.
4. Объектно-событийное и объектно-ориентированное программирование.		
17		Событие и сообщение. Виды событий. События от мыши и клавиатуры.
18		Программирование управления событиями. Организации главного меню, всплывающего меню и панели инструментов.
19		Обработка исключительных событий. Работа с диалогами в среде Gambas.
20		Основы визуального программирования. Компонент.
21		Иерархия компонентов.

### **5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)**

Формы контроля: устный (индивидуальный и фронтальный опрос), письменный (выполнение реферативной работы) и лабораторный контроль (решение индивидуальных задач).

Вопросы для устного контроля по темам дисциплины:

1. Параметры-переменные, параметры-значения, параметры-константы. Примеры.
2. Глобальные и локальные идентификаторы. Видимость объектов (идентификаторов). Примеры.
3. Область действия переменных и других идентификаторов.
4. Функции. Их отличие от процедур. Способ обращения к функции. Примеры.
5. Что такое комбинированный тип данных (запись)? Способы описания (примеры).
6. Способы обращения к компонентам записи.
7. Оператор присоединения и его использование.
8. Записи с вариантами. Примеры.

9. Понятие модуля. Разделы модуля.
10. Понятие указателя. Статические и динамические переменные.
11. Карта памяти. Динамическое распределение памяти. Создание и уничтожение динамических переменных.
12. Фрагментация динамической памяти. Освобождение динамической памяти.
13. Примеры использования динамической памяти.
14. Простые списки и действия с ними.
15. Стек. Процедуры обработки.
16. Очередь. Процедуры обработки.
17. Деревья. Формирование дерева. Добавление элемента в дерево. Способы обхода дерева. Исключение компонента из дерева.
18. Понятие об ООП. Отличие ООП подхода к программированию от структурного.
19. Основные понятия ООП. Примеры.
20. Инкапсуляция. Примеры.
21. Наследование. Примеры.
22. Полиморфизм. Примеры.
23. Инкапсулированные в классах поля, методы, свойства.
24. Разделы класса: `published`, `private`, `protected`, `public`.
25. Понятие о визуальном, событийно-управляемом программировании.

#### Темы рефератов:

1. Простые списки (коллекции, список переменного размера, класс `SimpleList`);
2. Неупорядоченные списки;
3. Связанные списки (добавление элементов, удаление элементов, уничтожение связанного списка, сигнальные метки, инкапсуляция связанных списков, доступ к ячейкам);
4. Разновидности связанных списков (циклические связанные списки, проблема циклических ссылок, двусвязанные списки, потоки);
5. Другие связанные структуры;
6. Псевдоуказатели;
7. Стеки и очереди;
8. Стеки (множественные стеки);
9. Очереди (циклические очереди, очереди на основе связанных списков, применение коллекций в качестве очередей, очереди с приоритетами, многопоточные очереди);
10. Деревья (представления деревьев, полные узлы, списки потомков, представление нумерацией связей, полные деревья, обход дерева);
11. Упорядоченные деревья (добавление элементов, удаление элементов, обход упорядоченных деревьев);
12. Деревья со ссылками (особенности работы);
13. Q-деревья (изменение количества элементов в узле, использование псевдоуказателей, восьмеричные деревья);
14. Сбалансированность дерева;

15. AVL-деревья (добавление узла, удаление узла);
16. B-деревья (производительность B-деревьев, вставка элементов, удаление элементов, разновидности B-деревьев);
17. Увеличение производительности B-деревьев (балансировка, вопросы, связанные с обращением к диску, база данных на основе B+дерева);
18. Деревья решений (поиск в деревьях игры: минимаксный поиск, оптимизация поиска);
19. Поиск нестандартных решений (метод ветвей и границ, эвристики);
20. Управляющие и контролируемый объекты;
21. Единственный объект. Порождающий объект.

## **6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)**

### **6.1. Паспорт фонда оценочных средств по дисциплине (модулю)**

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции* (или её части) / и ее формулировка – по желанию	наименование оценочного средства
1.	Процедуры и функции. Модули.	СПК-2	Лабораторная работа. Устный контроль.
2.	Организация динамических структур данных (абстрактных типов данных): списки, стеки, очереди, деревья.	СПК-2	Лабораторная работа. Реферат.
3.	Реализация абстракций данных методами объектно-ориентированного программирования.	СПК-2	Лабораторная работа. Устный контроль.
4.	Объектно-событийное и объектно-ориентированное программирование.	СПК-2	Лабораторная работа. Устный контроль.
5.	Зачет		

### **6.2. Типовые контрольные задания или иные материалы**

#### **6.2.1. Зачет**

1) типовые вопросы (задания)

Примеры задач по теме «Организация динамических структур данных (абстрактных типов данных): стек, очередь, двоичное дерево поиска».

#### **Списки**

##### **Вариант 1**

Написать программу для реверса списка. Например: список [1, 2, 3] преобразуется в список [3, 2, 1].

##### **Вариант 2**

Написать программу для получения значения n-го элемента списка. Например: в списке [three, one, two] второй элемент равен one.

### **Вариант 3**

Написать программу для удаления из списка всех элементов, равных 0. Например: список [1, 0, 2, 0, 0, 3] преобразуется в список [1, 2, 3].

### **Вариант 4**

Написать программу для циклического сдвига списка вправо на заданное число элементов. Например: список [6, 5, 4, 3, 2, 1], циклически сдвинутый вправо на 2 элемента, преобразуется в список [2, 1, 6, 5, 4, 3].

### **Вариант 5**

Написать программу для удаления из списка 2-ого, 4-ого и т.д. элементов. Например: список [6, 5, 4, 3, 2, 1] преобразуется в список [6, 4, 2].

### **Вариант 6**

Написать программу для замены в списке всех элементы, равные 0, на -1. Например: список [1, 0, 0] преобразуется в список [1, -1, -1].

### **Вариант 7**

Написать программу для перевода списка арабских чисел (от 1 до 10) в список римских чисел. Например: список [1, 2, 3] преобразуется в список ["I", "II", "III"].

### **Вариант 8**

Написать программу для подсчета количества определенных элементов в списке. Например: в списке [1, 2, 1, 3, 1] три единицы.

### **Вариант 9**

Написать программу для подсчета количества элементов списка без какого-либо указываемого элемента. Например: в списке [1, 2, 1, 3, 1] два элемента без учета единиц.

### **Вариант 10**

Написать программу для подсчета количества элементов списка, значения которых лежат в определенном диапазоне. Например: в списке [10, 20, 10, 30, 15] два элемента, значения которых больше 10 и меньше 30.

## **Деревья**

### **Вариант 1**

Написать программу для нахождения среднего арифметического листьев вершин бинарного дерева.

### **Вариант 2**

Написать программу для проверки упорядоченности бинарного дерева.

### **Вариант 3**

Вывести бинарное дерево на экран в виде дерева.

### **Вариант 4**

Написать программу для вычисления глубины бинарного дерева (глубина пустого дерева равна 0, глубина одноузлового дерева равна 1).

### **Вариант 5**

Написать программу для подсчета количества листьев вершин дерева, значения которых лежат в определенном диапазоне.

### **Вариант 6**

Написать программу для преобразования дерева в список.

### **Вариант 7**

Написать программу для нахождения среднего арифметического отрицательных узлов дерева.

#### **Вариант 8**

Написать программу для подсчета количества вершин бинарного дерева, значения которых не равны 0.

#### **Вариант 9**

Написать программу для нахождения среднего арифметического положительных узлов дерева.

#### **Вариант 10**

Написать программу для подсчета количества вершин бинарного дерева, значения которых равны 0.

2) критерии оценивания компетенций (результатов)

Результат оценивается в зависимости от правильности выполнения практического задания. Весьма важным в данном случае является временной фактор. Студент должен справляться с решением индивидуальной задачи по теме в рамках лабораторного занятия.

3) описание шкалы оценивания

Правильность выполнения практического задания оценивается по трехбалльной шкале: полностью правильно (2 балла), выполнено с недочетом (1 балл), выполнено полностью неверно (0 баллов).

#### **6.2.2 Наименование оценочного средства\*** (в соответствии с таблицей 6.1)

1) типовые задания (вопросы) - образец

Структура лабораторной работы представлена следующим образом:

- устный опрос по теме данного лабораторного занятия;
- выполнение лабораторной работы (индивидуальное, в микрогруппах).

Таким образом, на лабораторной работе занятии студент гарантированно получает оценку за опрос и за выполнение лабораторной работы.

*Лабораторная работа «Процедуры, функции. Разработка и вызов подпрограмм в среде Gambas».*

Gambas позволяет создавать пользовательские процедуры. Основной отличительной чертой пользовательских процедур является то, что они не связаны ни с каким событием и вызов их пользователь осуществляет по своему усмотрению. В виде пользовательской процедуры можно оформить любую подпрограмму и использовать ее в текущем проекте или сохранить на диске и использовать в других программах. Для создания пользовательской процедуры необходимо перейти к секции Главная, в окне программы ввести служебное слово **Sub** и **имя процедуры** и нажать клавишу Enter. После этого появится новая процедура.

Для вызова процедуры используются два способа: по имени процедуры и с помощью оператора Call. При вызове процедуры **по имени** после имени указываются без скобок передаваемые параметры:

ИмяПроцедуры аргумент1, аргумент2, ...

Передаваемые параметры по типам должны соответствовать типам переменных указанных в описании процедуры. В качестве передаваемых параметров могут использоваться строки символов, имена переменных или функций, возвращающих значения заданного типа.

При вызове процедуры с помощью оператор **Call** передаваемые

параметры заключаются в скобки:

```
Call ИмяПроцедуры (аргумент1, аргумент2,...)
```

Оператор Call используется, как правило, для вызова внешних процедур. Чтобы отличить вызов функции от вызова процедуры, рекомендуют не использовать оператор Call для вызова процедур (при вызове функций передаваемые параметры также заключаются в .скобки).

**Задание 1.** Разработайте приложение в среде Gambas, содержащую процедуру вычисления площади поверхности параллелепипеда.

Процедура, объявленная в секции главная формы

```
Sub SPOverch(a As Single, b As Single, h As Single, s As Single)
```

```
s = 2 * (a * b + (a + b) * h)
```

```
End Sub
```

```
Private Sub Form_Click()
```

```
‘ Вызывающая процедура
```

```
Dim x As Single, y As Single, z As Single, s As Single
```

```
x = Val(InputBox("Длина параллелепипеда"))
```

```
y = Val(InputBox("Ширина параллелепипеда "))
```

```
z = Val(InputBox("высота параллелепипеда "))
```

```
‘ вызов процедуры
```

```
SPOverch x, y, z, s
```

```
Print Str$(s)
```

```
End Sub
```

Наряду с процедурами можно создавать и пользовательские функции, синтаксис которых похож на синтаксис пользовательских.

Синтаксис функции пользователя:

```
Function ИмяФункции ( аргумент1 As тип, [, аргумент2 As тип ] ) As тип
```

```
< операторы >
```

```
ИмяФункции = результат
```

```
End Function.
```

В качестве аргументов могут быть константы, переменные или выражения. В конце процедуры записывается оператор **ИмяФункции = результат**, здесь **ИмяФункции** – простая переменная, имя которой совпадает с именем функции. Этим оператором имени функции передается вычисленное значение.

Функции пользователя используются так же, как и встроенные функции Visual Basic. То есть, они используются в выражениях справа от знака равно, могут включаться также в оператор Print:

```
< имя переменной > = ИмяФункции ( аргумент 1 [, аргумент 2, ...])
```

```
Print ИмяФункции
```

Допускается также использование функции в следующем формате:

```
ИмяФункции ( аргумент 1 [, аргумент 2, ...])
```

Внутри функции можно объявлять локальные переменные, указывая их тип: статические или динамические. Переменные уровня формы доступны всем функциям, подключенным к данному модулю или форме. Преждевременный

выход из функции осуществляется с помощью оператора **Exit Function**.

Вычислить значение Sin(x) с точностью 0.001. Математическая модель решения данной задачи представляется следующей формулой:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{x^{2i-1}}{(2i-1)!} + \dots = \sum_{i=1}^{\infty} \frac{x^{2i-1}}{(2i-1)!}$$

**Задание 2.** Разработайте приложение вычисления факториала с использованием функции. Входным параметром функции будет целое число одинарной длины n – факториал. Функция возвращает целое число двойной длины.

```
Public Function Factorial(n As Integer) As Double
    Dim i As Integer, F As Double
    F = 1
    For i = 1 To n
        F = F * i
    Next i
    Factorial = F
End Function
```

```
Private Sub Form_Click()
    Dim n As Integer, F As Double, k As Integer
    Dim x As Single, y As Single, s As Single
    x = Val(InputBox("Аргумент X"))
    e = Val(InputBox("Точность вычисления"))
    y = 1 + e: s = 0: k = 1
    While Abs(y) >= e
        y = (-1) ^ (k - 1) * x ^ (2 * k - 1) / Factorial(2 * k - 1)
        s = s + y
        k = k + 1
    Wend
    Print Str$(s)
End Sub
```

*Индивидуальные задачи для лабораторного практикума:*

### **Процедуры и функции**

1. Напишите программу, состоящую из трех процедур и основной программы. Первая процедура организует ввод двух целых чисел X и Y, вторая проверяет их сумму, третья выводит результат. Используйте эти процедуры в основной программе. Используйте X и Y как глобальные переменные.
2. Напишите программу вычисления площади поверхности и длины экватора на основе известного радиуса планет солнечной системы. Форму планет будем считать шаром. Вычисление площади поверхности и длины экватора оформите отдельными функциями.
3. Напишите программу поиска большего из четырех чисел с использованием подпрограммы поиска большего из двух чисел.
4. Даны координаты вершин многоугольника (x1, y1, x2, y2, ..., x10, y10). Напишите программу для вычисления его периметра (вычисление расстояния между вершинами оформить подпрограммой).

5. Напишите программу вычисления суммы:  $1! + 2! + 3! + \dots + n!$ , используя функцию вычисления факториала числа  $k$ .
6. Напишите программу для вычисления числа сочетаний из  $N$  по  $M$ . Число сочетаний определяется по формуле  $N!/(M!(N-M)!)$ , где  $N$  – количество элементов перебора. Используйте подпрограмму вычисления факториала.
7. Напишите программу для определения НОД трех натуральных чисел.
8. Даны действительные числа  $s, t$ . Составить программу вычисления выражения  $f(t, -2s, 1.17) + f(2.2, t, s-t)$ , где  $f(a, b, c) = (2a - b - \sin(c)) / (5 + |c|)$ .
9. Даны натуральные  $m$  и  $n$  ( $m < n$ ). Составить программу, сокращающую дробь  $m/n$ .
10. Напишите программу вычисления суммы квадратов простых чисел, лежащих в интервале  $(M, N)$ .
11. Напишите программу подсчета числа четных цифр, используемых в записи  $N$ -значного числа  $M$ .
12. Составьте программу вычисления суммы трехзначных чисел, в десятичной записи которых нет четных цифр.
13. Составьте программу вывода на экран всех натуральных чисел, не превосходящих  $N$  и делящихся на каждую из своих цифр.
14. Составьте программу нахождения наименьшего натурального  $N$ -значного числа  $X$  ( $X \geq 10$ ), равного утроенному произведению своих цифр.
15. Составьте программу подсчета числа всех натуральных чисел, меньших  $M$ , квадрат суммы цифр которых равен  $X$ .

### ***6.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций***

В конце изучения всех тем подводятся итоги работы студентов на практических занятиях путем суммирования всех заработанных баллов.

Поскольку студент выполняет различные виды работ, получает за них не только максимальное, но и минимальное количество баллов, то получаемый результат (сумма) целиком зависит от его активности в течение семестра. Выполняющий все задания студент значительно облегчает себе сдачу экзаменационного теста, поскольку набирает большое количество баллов предыдущими видами работ.

## 7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

### а) основная учебная литература:

1.Медведик, В.И. Практика программирования на языке Паскаль (задачи и решения) [Электронный ресурс] : учебное пособие. — Электрон. дан. — Москва : ДМК Пресс, 2013. — 590 с. — Режим доступа:<http://e.lanbook.com/book/58700>

2.Зыков, С. В. Программирование [Электронный ресурс]: учебник и практикум для академического бакалавриата / С. В. Зыков. — Электронные текстовые данные. - Москва: Издательство Юрайт, 2016.— 320 с. — (Бакалавр. Академический курс). — Режим доступа:

<https://biblio-online.ru/book/E10A680F-BAE2-4C4C-AE77-4BBF450B3EC9>

### б) дополнительная учебная литература:

1.Москвитин, А. А. Решение задач на компьютерах. В 2 ч. Ч. I. Постановка (спецификация) задач [Электронный ресурс] : учебное пособие / А. А. Москвитин. - Электрон. текстов. данные. - Москва ;Берлин : Директ-Медиа, 2015. - 165 с. - Режим доступа:<http://biblioclub.ru/index.php?page=book&id=273666>

2.Москвитин, А. А. Решение задач на компьютерах. В 2 ч. Ч. II. Разработка программных средств.[Электронный ресурс] : учебное пособие / А. А. Москвитин. – Электрон. текстов. данные. - Москва ;Берлин : Директ-Медиа, 2015. - 427 с. - Режим доступа:<http://biblioclub.ru/index.php?page=book&id=273667>

## 8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

Электронно-библиотечные системы: ЭБС Издательства Лань <http://e.lanbook.com/>, ЭБС «znanium.com» <http://znanium.com>, ЭБС «Университетская библиотека online» <http://biblioclub.ru>

Газзаев Е. Опыт в программировании на Gambas // <http://ibone.org.ua/gambas/gambas-file-management-1.htm>

Черный В. Программирование в Gambas // <http://freeschool.altlinux.ru/?p=4234>

## 9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид учебных занятий	Организация деятельности студента
Лабораторная работа	Вузовская подготовка специалистов должна обеспечивать приобретение ими не только знаний, но и умений использовать полученные знания на практике. Это требование и положено в основу целей и методов проведения лабораторных работ по вышеуказанной учебной

		дисциплине. Лабораторные работы предлагаются в соответствии с рабочей программой в рамках каждой темы.
Подготовка к зачету	к	Подготовка к зачету предполагает изучение рекомендуемой литературы и других источников, конспектов лекций, повторение материалов практических занятий.

**10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

1. Применяется системное и прикладное программное обеспечение при выполнении лабораторных работ.
2. Используются электронные ресурсы и ресурсы Интернет для подготовки к занятиям;
3. Консультирование студентов и контроль выполнения лабораторных работ осуществляется посредством электронной почты.

**11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

Для проведения практических занятий необходима аудитория на 20 мест.

Для использования электронных ресурсов в рамках проведения лабораторных занятий необходима персональная компьютерная техника с доступом в Интернет.

При выполнении лабораторных работ применяется операционная система Linux и объектно-ориентированная среда разработки программ Gambas.

**12. Иные сведения и (или) материалы**

**12.1. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**

Для обеспечения образования инвалидов и обучающихся с ограниченными возможностями здоровья разрабатывается адаптированная образовательная программа, индивидуальный учебный план с учетом особенностей их психофизического развития и состояния здоровья.

Обучение обучающихся с ограниченными возможностями здоровья осуществляется на основе образовательных программ, адаптированных для обучения указанных обучающихся.

Обучение по образовательной программе инвалидов и обучающихся с ограниченными возможностями здоровья осуществляется факультетом с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Университетом создаются специальные условия для получения высшего образования по образовательным программам обучающихся с ограниченными возможностями здоровья.

## 12.2 Занятия, проводимые в интерактивных формах

№ п/п	Раздел, тема дисциплины	Объем аудиторной работы в интерактивных формах по видам занятий (час.)			Формы работы
		Лекц.	Практич	Лабор	
1	Процедуры и функции. Модули.	2		2	Дискуссия, работа в малых группах
2	Организация динамических структур данных (абстрактных типов данных): списки, стеки, очереди, деревья.	2		2	Дискуссия, работа в малых группах
3	2. Объектно- ориентированное программирование			4	Работа в малых группах
4	Реализация абстракций данных методами объектно- ориентированного программирования.			4	Работа в малых группах
	<b>ИТОГО по дисциплине:</b>	<b>4</b>		<b>12</b>	

Составитель (и): Буяковская И.А., доцент

(фамилия, инициалы и должность преподавателя (ей))

Макет рабочей программы дисциплины (модуля) разработан в соответствии с приказом Минобрнауки России от 19.12.2013 № 1367, одобрен научно-методическим советом (протокол № 8 от 09.04.2014 г.) и утвержден приказом ректора от 23.04.2014 № 224/10..

Макет обновлён с поправками в части подписей на титульной странице, п.3 добавлена строка для указания часов, проводимых в активной и интерактивной формах обучения, добавлен п. 12.1 Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья (протокол НМС № 6 от 15.04.2015 г.), утвержден приказом ректора.