

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Приднестровский государственный университет
имени Т.Г. Шевченко»

Рыбницкий филиал

Кафедра информатики и программной инженерии

«УТВЕРЖДАЮ»

Директор филиала ПГУ им. Т.Г. Шевченко в г. Рыбнице
профессор  И.А. Павлинов

« 15 » 09 2017 г.

РАБОЧАЯ ПРОГРАММА

на 2017/2018 учебный год

УЧЕБНОЙ ДИСЦИПЛИНЫ

«ОСНОВЫ ПРОГРАММИРОВАНИЯ»

Направление подготовки:

09.03.04 «Программная инженерия»

Профиль подготовки:

«Разработка программно-информационных систем»

Квалификация (степень) выпускника:

бакалавр

Форма обучения:

очная

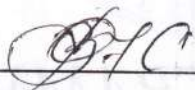
Рыбница 2017

Рабочая программа дисциплины «*Основы программирования*» / сост.:
О.В. Сташкова, Е.С. Гарбузняк, Н.С. Чебан. – Рыбница: филиал ПГУ им. Т.Г. Шевченко в
г. Рыбнице, 2017. – 33 с.

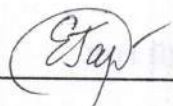
**РАБОЧАЯ ПРОГРАММА ПРЕДНАЗНАЧЕНА ДЛЯ ПРЕПОДАВАНИЯ
ДИСЦИПЛИНЫ БАЗОВОЙ ЧАСТИ БЛОКА ДИСЦИПЛИН (МОДУЛЕЙ) СТУДЕНТАМ
ОЧНОЙ ФОРМЫ ОБУЧЕНИЯ ПО НАПРАВЛЕНИЮ ПОДГОТОВКИ 09.03.04 –
«ПРОГРАММНАЯ ИНЖЕНЕРИЯ».**

Рабочая программа составлена с учетом Федерального Государственного
образовательного стандарта высшего образования по направлению подготовки 09.03.04 –
«Программная инженерия», утвержденного приказом Министерства образования и науки
Российской Федерации от 12 марта 2015 г. № 229.

Составители:



О.В. Сташкова, ст. преподаватель



Е.С. Гарбузняк, ст. преподаватель



Н.С. Чебан, преподаватель

1. Цели и задачи освоения дисциплины

Целью освоения дисциплины «Основы программирования» является изучение теоретических и практических основ программирования.

Задачами дисциплины являются:

- получение представления об основах программирования;
- приобретение знаний о принципах организации, структурах технических и программных средств, используемых в программировании;
- приобретение навыков практического использования инструментальных средств программирования в информационных системах.

2. Место дисциплины в структуре ООП ВО

Дисциплина «Основы программирования» относится к базовой части (Б1.Б.11) блока дисциплин (модулей) подготовки студентов по направлению 09.03.04 – «Программная инженерия».

Изучаемая дисциплина связана со следующим дисциплинами, изучающимися студентами параллельно или сразу после дисциплины «Основы программирования»: «Иностранный язык», «Теоретическая информатика», «Введение в алгоритмы», «Типы и структуры данных», «Объектно-ориентированное программирование».

Поскольку дисциплина является вводной в проблему информационных технологий, то специальных требований к «входным» знаниям, умениям и готовностям обучающегося, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин не предъявляется.

3. Требования к результатам освоения дисциплины

Изучение дисциплины направлено на формирование следующих компетенций:

Код компетенции	Формулировка компетенции
ОПК-1	Владение основными концепциями, принципами, теориями и фактами, связанными с информатикой
ОПК-3	Готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов
ОПК-4	Способность осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий
ПК-1	Готовность применять основные методы и инструменты разработки программного обеспечения
ПК-12	Способность к формализации в своей предметной области с учетом ограничений используемых методов исследования
ПК-15	Способность готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях
ПК-16	Способность формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта

ПК-17	Способность выполнить начальную степень трудности, рисков, затрат и сформировать рабочий график
ПК-21	Владение навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации

В результате освоения дисциплины студент должен:

3.1. Знать:

- формальные методы, технологии и инструменты разработки программного продукта;
- концепции эволюционного развития программного обеспечения;
- концепции и реализации программных процессов.

3.2. Уметь:

- устанавливать, тестировать, испытывать и использовать программные средства;
- конструировать программное обеспечение;
- разрабатывать основные программные документы;
- работать с современными системами программирования.

3.3. Владеть: навыками разработки и отладки программ на языке высокого уровня.

4. Структура и содержание дисциплины

Рабочая программа учебной дисциплины рассчитана на 2 семестра. Трудоемкость дисциплины на 2 семестра составляет 12 зачетных единиц, 432 часа. В том числе 54 часа отводится на лекционные занятия, 90 часов – на лабораторные занятия, 54 часа – на практические занятия, 162 часа – на самостоятельную работу.

4.1. Распределение трудоемкости в з.е./часах по видам аудиторной и самостоятельной работы студентов по семестрам

Семестр	Количество часов						Форма итогового контроля
	Трудоем- кость, з.е./часы	В том числе					
		Аудиторных				Самост. работы	
		Всего	Лекций	Лаб. раб.	Практич. зан.		
I	5/180	72	18	36	18	72	экзамен 36
II	7/252	126	36	54	36	90	экзамен, курсовая работа 36
Итого:	12/432	198	54	90	54	162	72

4.2. Распределение видов учебной работы и их трудоемкости по разделам дисциплины

№ раз- дела	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Внеауд. работа (СР)
			Л	ПЗ	ЛР	
1	Введение. Синтаксис языка.	36	6	4	8	18
2	Типы данных. Операции над данными.	48	6	4	12	26
3	Структурное программирование.	50	6	8	14	22

№ раз-дела	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Внеауд. работа (СР)
			Л	ПЗ	ЛР	
4	Модульное программирование.	46	8	8	10	20
5	Составные типы данных.	128	16	24	38	50
6	Стандартные библиотеки Си++	44	6	6	6	26
7	Введение в объектно-ориентированное программирование.	8	6	–	2	–
Всего:		360	54	54	90	162

4.3. Тематический план по видам учебной деятельности

Лекции

№ п/п	Номер раздела дисциплины	Объем часов	Тема лекции	Учебно-наглядные пособия
I СЕМЕСТР				
1	1	2	Программирование на языках высокого уровня (на примере Си/С++). Язык Си: история, первоначальная область применения (системное программирование). Принцип построения: компилируемые конструкции и интерпретируемые средства (библиотека стандартных функций). Раздельная трансляция, компилятор и редактор связей.	–
2	1	2	Ввод информации. Управляющие коды. Отображение специальных символов на экране монитора. Форматированный вывод. Организация ввода данных. Входной поток. Организация диалога. Комментарии.	Компьютерные презентации
3	1	2	Алфавит языка. Способы описания синтаксиса языка. Определение понятия «идентификатор». Служебные слова. Комментарии. Типы данных. Имена и объявления. Описание данных, литералы. Назначение стандартных заголовочных файлов. Компоновка программы.	Методическое пособие
4	2	2	Целые типы данных: int, short, long, char. Кодовый формат. Тип целых констант. Арифметические операции для целых операндов. Понятие преобразования данных. Явное преобразование типа. Правила преобразования операндов в процессе вычислений.	Компьютерные презентации
5	2	2	Плавающие типы данных: float, double. Кодовый формат. Описание данных, литералы. Тип констант с плавающей точкой. Арифметические операции для вещественных операндов. Математические функции стандартной библиотеки Си из объектных модулей и библиотек.	Компьютерные презентации

6	2	2	Оператор-выражение. Операции отношения. Результат вычисления отношений. Представление булевских значений «ложь», «истина» в Си. Логические операции: !, , &&. Операции простого и составного присваивания. Приоритеты операций.	Методическое пособие
7	3	2	Структурное программирование. Операторы ветвления: if и switch. Оператор if, синтаксис, выполнение оператора. Операции отношения. Составной оператор. Сложные условия и логические операции: !, , &&. Вложение операторов if, оператор if ... else if ... else. Оператор switch, выполнение, использование оператора break.	Компьютерные презентации
8	3	2	Структурное программирование. Операторы цикла с предусловием (while) и постусловием (do-while). Примеры применения циклов. Изменение хода выполнения цикла с помощью операторов break и continue.	—
9	4	2	Модульное программирование. Функции. Рациональные размер и количество параметров функции. Пример функции. Аргументы и параметры. Передача аргументов по значению и по ссылке. Прототипы функций. Преобразование аргументов в точке вызова. Оператор return.	Методическое пособие
Итого:		18		
II СЕМЕСТР				
1	4	2	Область действия переменных. Локальные и глобальные переменные. Видимость переменных. Время жизни переменной. Модификаторы переменных. Изменяющиеся переменные.	Методическое пособие
2	4	2	Рекурсивные подпрограммы. Понятие о рекурсии. Сравнение рекуррентности и рекурсии. Прямой и обратный ход рекурсии. Точка останова работы рекурсивного алгоритма. Глубина рекурсии.	—
3	4	2	Обеспечение корректной стыковки модулей. Создание библиотек функций.	—
4	4	2	Создание заголовочных файлов.	—
5	5	2	Массивы, примеры использования. Обращение к элементу массива. Работа с элементами массива. Многомерные массивы.	Компьютерные презентации
6	5	2	Сортировка массивов. Пузырьковая сортировка. Сортировка выбором. Сортировка вставкой.	Компьютерные презентации
7	5	2	Указатели. Указатели и массивы. Указатели и функции.	Методическое пособие
8	5	2	Тип данных строки. Функции работы со строками. Таблица ASCII.	Методическое пособие

9	5	2	Структура как тип данных. Поле структуры. Обращение к полям структуры. Размещение структур в памяти. Примеры использования.	Методическое пособие
10	5	2	Типы, определяемые пользователем. Объединения. Перечисления. Переименование типов данных.	Методическое пособие
11	5	2	Тип данных файлы. Работа с файлами. Поточковый ввод/вывод данных.	Методическое пособие
12	5	2	Понятие о динамических данных. Линейные списки. Организация данных в виде стека, очереди, дерева.	Компьютерные презентации
13	6	2	Библиотека <string.h>.	—
14	6	2	Обработка исключительных ситуаций.	—
15	6	2	Работа с графикой. Функции работы с графикой.	Компьютерные презентации
16	7	2	Ведение в объектно-ориентированное программирование. Объекты и классы.	—
17	7	2	Среда разработки Borland C++ Builder. Основные компоненты. Свойства. События. Методы.	Компьютерные презентации
18	7	2	Создание Windows-приложений в среде Borland C++ Builder. Менеджер проектов.	—
Итого:		36		
Всего:		54		

Практические (семинарские) занятия

№ п/п	Номер раздела дисциплины	Объем часов	Тема практического занятия	Учебно-наглядные пособия
I СЕМЕСТР				
1	1	2	Типы данных C++. Литералы.	Раздаточный материал
2	1	2	Операторы. Преобразование типов в операторах присваивания. Выражения.	Раздаточный материал
3	2	2	Арифметические операции для целых и вещественных типов данных.	Карточки с заданиями
4	2	2	Математические функции стандартной библиотеки Си из объектных модулей и библиотек.	Карточки с заданиями
5	3	2	Операторы ветвления: if и switch.	Карточки с заданиями
6	3	2	Оператор повторений for.	Карточки с заданиями
7	3	2	Оператор повторений while. Оператор повторений do ... while.	Карточки с заданиями
8	3	2	Инструкции управления.	Раздаточный материал
9	4	2	Основы использования функций.	Раздаточный материал
Итого:		18		

II СЕМЕСТР				
1	4	2	Создание пользовательских подпрограмм.	Карточки с заданиями
2	4	2	Использование рекурсии для решения задач.	Раздаточный материал
3	4	2	Решение рекуррентных уравнений	Карточки с заданиями
4	5	2	Работа с одномерными массивами.	Карточки с заданиями
5	5	2	Работа с многомерными массивами.	Карточки с заданиями
6	5	2	Решение задач с применением сортировок массива.	Раздаточный материал
7	5	2	Символьный тип данных.	Карточки с заданиями
8	5	2	Работа со строками. Использование строковых функций.	Раздаточный материал
9	5	2	Использование типа данных структуры.	Карточки с заданиями
10	5	2	Использование типа данных объединения.	Карточки с заданиями
11	5	2	Использование перечисляемого типа данных.	Карточки с заданиями
12	5	2	Переименование типов данных.	Карточки с заданиями
13	5	2	Потоковый ввод-вывод.	Раздаточный материал
14	5	2	Форматированный ввод-вывод.	Раздаточный материал
15	5	2	Прямой доступ к файлам.	Раздаточный материал
16	6	2	Использование функций для работы со строками из стандартной библиотеки <string.h>.	Карточки с заданиями
17	6	2	Работа с графикой. Функции работы с графикой.	Раздаточный материал
18	6	2	Работа с графическими примитивами.	Раздаточный материал
Итого:		36		
Всего:		54		

Лабораторные работы

№ п/п	Номер раздела дисциплины	Объем часов	Тема лабораторного занятия	Наименование лаборатории	Учебно-наглядные пособия
I СЕМЕСТР					
1	1	2	Инструментарий среды.	Компьютерная аудитория	Практикум
2	1	2	Элементы языка. Структура программы. Состав языка.	Компьютерная аудитория	Практикум

3	2	2	Типы данных в Си. Выражения, операции, операнды. Простые операторы.	Компьютерная аудитория	Практикум
4	1	2	Ввод/вывод информации. Форматная строка.	Компьютерная аудитория	Практикум
5	1	2	Оператор присваивания, модифицированный оператор присваивания.	Компьютерная аудитория	Практикум
6	3	2	Пользовательские подпрограммы. Функциональное программирование.	Компьютерная аудитория	Практикум
7	2	2	Группы целых типов.	Компьютерная аудитория	Практикум
8	2	2	Операции для работы с целыми величинами.	Компьютерная аудитория	Практикум
9	2	2	Группы вещественных типов.	Компьютерная аудитория	Практикум
10	2	2	Операции для работы с вещественными величинами.	Компьютерная аудитория	Практикум
11	2	2	Группы логических типов. Логические операции в C++.	Компьютерная аудитория	Практикум
12	3	2	Тернарный оператор. Условный оператор switch.	Компьютерная аудитория	Практикум
13	3	2	Условный оператор. Вложение операторов if, оператор if ... else if ... else.	Компьютерная аудитория	Практикум
14	3	2	Оператор повторений с предусловием.	Компьютерная аудитория	Практикум
15	3	2	Оператор повторений с постусловием.	Компьютерная аудитория	Практикум
16	3	2	Оператор повторений со счётчиком.	Компьютерная аудитория	Практикум
17	3	2	Изменение хода выполнения цикла с помощью операторов break и continue.	Компьютерная аудитория	Практикум
18	4	2	Оператор return.	Компьютерная аудитория	Практикум
Итого:		36			
II СЕМЕСТР					
1	4	2	Реккурентные соотношения. Использование рекурсивных программ.	Компьютерная аудитория	Практикум
2	4	2	Нахождение числа Фибоначчи через рекуррентное соотношение.	Компьютерная аудитория	Практикум
3	4	2	Решение задач с использованием прямого и обратного хода рекурсии.	Компьютерная аудитория	Практикум
4	4	2	Решение рекуррентных уравнений с использованием рекурсии.	Компьютерная аудитория	Практикум
5	5	2	Использование одномерных массивов.	Компьютерная аудитория	Практикум

6	5	2	Реализация алгоритмов работы с одномерными массивами.	Компьютерная аудитория	Практикум
7	5	2	Работа с многомерными массивами.	Компьютерная аудитория	Практикум
8	5	2	Реализация алгоритмов работы с многомерными массивами.	Компьютерная аудитория	Практикум
9	5	2	Реализация пузырьковой сортировки.	Компьютерная аудитория	Практикум
10	5	2	Реализация сортировки выбором.	Компьютерная аудитория	Практикум
11	5	2	Реализация сортировки вставкой.	Компьютерная аудитория	Практикум
12	5	2	Использование строк в языке Си++.	Компьютерная аудитория	Практикум
13	5	2	Функции работы со строками.	Компьютерная аудитория	Практикум
14	5	2	Реализация алгоритмов работы со строками.	Компьютерная аудитория	Практикум
15	5	2	Работа со структурами на языке Си++.	Компьютерная аудитория	Практикум
16	5	2	Обращение к полям структуры. Оператор «.».	Компьютерная аудитория	Практикум
17	5	2	Использование объединений в программах на Си++.	Компьютерная аудитория	Практикум
18	5	2	Использование перечислений в программах на Си++.	Компьютерная аудитория	Практикум
19	5	2	Переименование типов данных. Использование оператора typedef.	Компьютерная аудитория	Практикум
20	5	2	Работа с файлами средствами библиотеки <stdio.h>	Компьютерная аудитория	Практикум
21	5	2	Форматированный ввод\вывод файлов.	Компьютерная аудитория	Практикум
22	5	2	Потоковый ввод\вывод файлов.	Компьютерная аудитория	Практикум
23	5	2	Работа с бинарными файлами.	Компьютерная аудитория	Практикум
24	6	2	Работа со строками при помощи библиотеки <string.h>.	Компьютерная аудитория	Практикум
25	6	2	Реализация консольного текстового редактора.	Компьютерная аудитория	Практикум
26	6	2	Инициализация графики. Выход из графического режима. Рисование графических примитивов.	Компьютерная аудитория	Практикум
27	7	2	Создание простого приложения в среде Borland C++ Builder.	Компьютерная аудитория	Практикум
Итого:		54			
Всего:		90			

Самостоятельная работа студента

Раздел дисциплины	№ п/п	Тема и вид СРС	Трудоемкость (в часах)
1	1	Язык Си: история, первоначальная область применения. <i>Подготовка сообщений.</i>	2
	2	Изучение инструментальной среды Turbo C. <i>Практическая работа.</i>	2
	3	Сравнительный анализ операторов ввода и вывода данных. <i>Конспектирование.</i>	4
	4	Решение задач с использованием различных операторов ввода и вывода.	4
	5	Синтаксис языка Си. <i>Работа с литературой.</i>	2
	6	Назначение стандартных заголовочных файлов. <i>Работа с литературой.</i>	2
	7	Назначение стандартных заголовочных файлов. <i>Конспектирование.</i>	2
2	8	Целые типы данных. <i>Работа с литературой.</i>	2
	9	Решение задач с использованием целых типов данных.	4
	10	Вещественные типы данных. <i>Работа с литературой.</i>	2
	11	Решение задач с использованием вещественных типов данных.	4
	12	Решение задач с использованием математических функций.	4
	13	Операции отношения. Логические операции. <i>Работа с литературой.</i>	2
	14	Решение задач с использованием логических операций.	4
3	15	Сравнительный анализ типов данных. <i>Конспектирование.</i>	4
	16	Операторы ветвления. <i>Работа с литературой.</i>	2
	17	Решение задач с использованием оператора ветвления if.	4
	18	Решение задач с использованием оператора ветвления switch.	4
	19	Операторы цикла. <i>Работа с литературой.</i>	2
	20	Решение задач с использованием оператора цикла со счетчиком.	2
	21	Решение задач с использованием оператора цикла с предусловием.	2
	22	Решение задач с использованием оператора цикла с постусловием.	2
	23	Сравнительный анализ операторов цикла. <i>Конспектирование.</i>	2
4	24	Решение задач с использованием операторов цикла	2
	25	Модульное программирование. <i>Работа с литературой.</i>	2
	26	Решение задач с использованием пользовательских функций.	4
	27	Изучение рекурсивных алгоритмов. <i>Работа с литературой.</i>	4

	28	Косвенная рекурсия. <i>Работа с литературой.</i>	2
	29	Создание заголовочных файлов.	4
	30	Область видимости переменных. Сравнительная характеристика. <i>Работа с литературой.</i>	2
	31	Время жизни переменных. <i>Работа с литературой.</i>	2
5	32	Сортировки массива. <i>Подготовка сообщений.</i>	4
	33	Решение задач с применением пирамидальной и бинарной сортировок	2
	34	Сравнительная характеристика сортировки Шелла и сортировки слиянием. <i>Работа с литературой.</i>	4
	35	Реализация сортировок над массивами.	2
	36	Реализация алгоритмов поиска в массиве.	2
	37	Кодировка ASCII и Unicode. <i>Работа с литературой.</i>	2
	38	Реализация алгоритма поиска подстроки.	2
	39	Работа со строками в Unicode.	2
	40	Реализация ввода/вывода структуры.	2
	41	Выравнивание структур памяти. <i>Работа с литературой.</i>	2
	42	Использование перечислений в программах. <i>Работа с литературой.</i>	4
	43	Использование оператора typedef. <i>Работа с литературой.</i>	2
	44	Использование объединений в программах. <i>Работа с литературой.</i>	2
	45	Сравнительная характеристика: различие объединений и структур. <i>Работа с литературой.</i>	2
	46	Работа с файлами средствами языка Си++. <i>Работа с литературой.</i>	4
	47	Сравнительная характеристика: различие режимов открытия файлов. <i>Работа с литературой.</i>	2
	48	Использование операций seek и feof. <i>Работа с литературой.</i>	2
	49	Сохранение и чтение структур из файлов. <i>Работа с литературой.</i>	2
	50	Работа с файлами как с потоками. <i>Работа с литературой.</i>	2
	51	Использование модификаторов при форматированном вводе/выводе из файлов. <i>Работа с литературой.</i>	4
6	52	Реализация основных алгоритмов работы с типом string	4
	53	Сравнительная характеристика: различие типа string и стандартных строк. <i>Работа с литературой.</i>	2
	54	Изучение функций работы с типом string. <i>Работа с литературой.</i>	4
	55	Реализация вывода текста в графическом режиме.	4
	56	Создание графического меню.	6
	57	Изменение цвета и заливки.	2
	58	Система координат в графическом режиме. <i>Работа с литературой.</i>	4
Всего:			162

5. Примерная тематика курсовых проектов (работ)

1. Разработка электронного словаря с возможностью подключения к базе данных.
2. Программы для построения графиков математических функций.
3. Разработка простейшего почтового клиента.
4. Разработка калькулятора логических функций.
5. Разработка интерактивной логической игры «Квест».
6. Разработка программы «Слайд шоу».
7. Разработка интерактивной развивающей игры «Найди отличия».
8. Разработка графического редактора со сменными фонами «Раскраска».
9. Разработка игрового приложения «Составь слово».
10. Разработка приложения «Организатор».
11. Программа, моделирующая работу стрелочных часов с кукушкой.
12. Разработка информационно-поисковой системы.
13. Разработка клавиатурного тренажера.
14. Разработка программы «Калькулятор квартплаты».
15. Программа автоматического написания характеристик.
16. Разработка графического редактора для геометрического отображения графов.
17. Разработка обучающе-контролирующей программы по системам счисления.

6. Образовательные технологии

Семестр	Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
I	Л	Проблемная лекция с использованием электронных образовательных ресурсов Лекция-визуализация.	8
	ПР	Изучение и закрепление нового материала (работа с наглядными пособиями, видео- и аудиоматериалами)	6
	ЛР	Работа в малых группах. Творческие задания.	10
Итого:			24
II	Л	Проблемная лекция с использованием электронных образовательных ресурсов Лекция-визуализация.	10
	ПР	Изучение и закрепление нового материала (работа с наглядными пособиями, видео- и аудиоматериалами)	8
	ЛР	Работа в малых группах. Творческие задания.	20
Итого:			38
Всего			64

7. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Для оценки качества усвоения курса используются следующие формы контроля:

- **текущий** – контроль выполнения лабораторных и индивидуальных заданий;
- **рубежный** – написание модульных контрольных работ;

– **итоговый** осуществляется посредством тестирования, экзамена, защиты курсовой работы.

Контроль самостоятельной работы студентов осуществляется с помощью ответов на практических и лабораторных занятиях, консультациях, ответов на тестирование.

**Образец теста для проведения итогового контроля
по итогам освоения дисциплины, а также для контроля самостоятельной работы
студента**

Структура педагогических измерительных материалов

Номер задания	Наименование темы задания
ДЕ 1. Введение. Синтаксис языка (критерий освоения ДЕ: не менее 5 правильно выполненных заданий)	
1-5	Среда Borland C.
6-8	Заголовочные файлы. Назначение стандартных заголовочных файлов.
9-12	Ввод и вывод информации. Форматированный вывод.
13,14	Алфавит языка. Способы описания синтаксиса языка. Определение понятия «идентификатор». Служебные слова.
ДЕ 2. Типы данных. Операции над данными (критерий освоения ДЕ: не менее 6 правильно выполненных заданий)	
15-18	Операции отношения. Логические операции. Приоритет операций.
19,20	Вещественные типы данных. Арифметические операции для вещественных операндов. Математические функции стандартной библиотеки Си.
21-32	Целые типы данных. Арифметические операции для целых операндов.
ДЕ 3. Структурное программирование (критерий освоения ДЕ: не менее 11 правильно выполненных заданий)	
33-48	Операторы ветвления: if и switch.
49-65	Операторы цикла: с предусловием, с постусловием и со счетчиком.
66-68	Изменение хода выполнения цикла с помощью операторов break и continue.
ДЕ 4. Модульное программирование (критерий освоения ДЕ: не менее 4 правильно выполненного задания)	
69-80	Функции. Аргументы и параметры. Прототипы функций.

Указания: Напишите Вашу фамилию, номер группы и дату. Для ответа на вопрос с выбором варианта ответа достаточно написать номер вопроса и рядом букву, обозначающую правильный вариант из предложенных в тексте ответов на вопрос. Если Вы считаете правильными несколько вариантов ответов, то запишите через запятую соответствующие литеры букв. Ответ считается верным, если отмечены все правильные варианты ответов.

Время выполнения – 90 минут.

ДЕ 1. Введение. Синтаксис языка – 14

1. Функциональная клавиша, позволяющая перейти к следующему окну:

- a) F3;
- b) F4;
- c) F5;
- d) F6.

2. Комбинация клавиш, позволяющая зайти в режим Window/Size/Move:

- a) Shift+F5;
- b) Ctrl+F5;
- c) Shift+F6;
- d) Ctrl+F6.

3. Комбинация клавиш, позволяющая откомпилировать программу:

- a) Alt+F9;
- b) Ctrl+F9;
- c) Ctrl+F5;
- d) Alt+F5.

4. Комбинация клавиш, позволяющая стереть строку на которой располагается курсор:

- a) Ctrl+K;
- b) Ctrl+Y;
- c) Alt+K;
- d) Alt+Y.

5. Комбинация клавиш Shift+Del позволяет:

- a) скопировать;
- b) вырезать;
- c) вставить;
- d) очистить.

6. Директива, позволяющая включить в текст программ другой текстовый файл:

- a) define;
- b) include;
- c) pragma;
- d) void.

7. Правильными высказываниями о директивах препроцессора являются:

- a) в конце директивы должна стоять точка с запятой;
- b) директива начинается со знака #.

8. Функции printf и scanf описываются в библиотечном файле:

- a) iostream.h;
- b) stdio.h;
- c) conio.h;
- d) math.h.

9. Для комментариев могут использоваться символы:

- a) /* comment */;
- b) { comment };
- c) (* comment *).

10. Укажите строки, не содержащие ошибок синтаксиса:

- a) printf("Hello, World!\n");
- b) printf("Hello, World!\n");
- c) printf(%d Hello, \n Hello, World!?);
- d) printf("Hello,\n World!\n").

11. В результате выполнения операторов

```
int n; n=5;
printf("%d+%d=%d", n, n, n+n);
```

будет напечатано:

- a) 5 + 5=10;
- b) 5;
- c) 10;
- d) 5+5.

12. Укажите, будет ли работать следующая программа:

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
printf("Hello, World!\n");
```

```
}
```

- a) программа скомпилируется и отработает, но код завершения будет не определен;
- b) программа не скомпилируется из-за синтаксической ошибки;
- c) программа не скомпилируется, т.к. не указан возвращаемый функцией тип.

13. Элементами алгоритмического языка C являются:

- a) идентификаторы;
- b) ключевые слова;
- c) разделители;
- d) римские цифры.

14. На языке Си правильно написаны имена:

- a) Name;
- b) name_;
- c) +name;
- d) 3name.

ДЕ2 Типы данных. Операции над данными – 18

15. Унарными операциями являются:

- a) &, !, ++;
- b) &, ~, ^;
- c) *, +, %;
- d) >, --, |.

16. Бинарные арифметические операции используются в выражениях:

- a) X + Y;
- b) c % d + 2;
- c) xx * Y;
- d) xx++.

17. Укажите правильный приоритет операций.

- a) (), <<>, +, !;
- b) !, <, &&, ?;
- c) &, *, ||, ();
- d) (), ^, *, +.

18. Если (Выражение1) в (Выражение1) ? (Выражение2): (Выражение3) будет истинно:

- a) выполнится выражение 2;
- b) выполнится выражение 3;

- c) выполнится выражение 2, а затем выражение 3;
d) выполнится выражение 3, а затем выражение 2.

19. Математическая функция, вычисляющая $\ln(x)$, выглядит:

- a) $\ln(x)$;
b) $\log(x)$;
c) $\lg(x)$;
d) $\log_e(x)$.

20. Укажите соответствие математической функции и ее назначения:

- | | |
|---------------------|--------------------------------------|
| a) $\text{abs}(x)$ | 1) Вычисляет «e» в степени x; |
| b) $\text{sqrt}(x)$ | 2) Возвращает модуль целого числа; |
| c) $\text{exp}(x)$ | 3) Возвращает модуль дробного числа; |
| d) $\text{fabs}(x)$ | 4) Вычисляет квадратный корень. |

21. После выполнения операторов $x = 5$; $x += --x$;
x будет равен:

- a) 4;
b) 5;
c) 8;
d) 6.

22. В случае $\text{int } t = 11/2$;
переменная t примет значение:

- a) 5;
b) 5.4999;
c) 5.5.

23. В случае $\text{int } c$; $c = 11\%3$;
переменная c примет значение:

- a) 0.2;
b) 2;
c) 3.2.

24. `unsigned char` принимает диапазон значений:

- a) $[-128..127]$;
b) $[0..255]$;
c) $(0..255)$;
d) $(-128..127)$.

25. Результатом вычисления $5 << 3$ будет:

- a) 40;
b) 3;
c) 0;
d) 12.

26. Операция, позволяющая определить количество байт, которые занимает переменная t:

- a) `(double) t`;
b) `sizeof (t)`;
c) `sizeof t`.

27. Какой(ие) из вариантов позволяет преобразовать переменную `t` типа `int` к вещественному типу?

- a) `(double) t`;
- b) `(float) t`;
- c) `(long int) t`.

28. После выполнения фрагмента

```
int y=10;
```

```
( y > 15) ? ( y *= 10) : ( y /= 7);
```

переменная `y` примет значение:

- a) 1;
- b) 1.42;
- c) 100.

29. В результате работы программы

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void main (void)
```

```
{
```

```
    short n;
```

```
    n=32770;
```

```
    cout<<"n"<<n;
```

```
    getch();
```

```
}
```

на экран будет выведено:

- a) 32770;
- b) -32766;
- c) 2;
- d) выведет сообщение об ошибке.

30. Запишите результат работы программы:

```
#include<stdio.h>
```

```
main(void)
```

```
{ int x=5;
```

```
  printf("x=%d \n",++x);}
```

- a) 5;
- b) 4;
- c) 6;
- d) 7.

31. Результат выполнения следующего фрагмента кода: `cout << 22 / 5 * 3`;

- a) 1.47;
- b) 1;
- c) 12;
- d) 13.2.

32. Ключевое слово, указывающее, что целая переменная не может принимать отрицательные значения:

- a) `unsigned`;
- b) нет такого зарезервированного слова;
- c) `positive`;
- d) `long`.

ДЕ 3. Структурное программирование – 36

33. Если условие условного оператора ложное, то:

- a) выполняется следующий оператор, сразу после оператора if;
- b) программа завершает работу;
- c) выполняется тело условного оператора.

34. Логическим оператором И является:

- a) &&;
- b) |;
- c) &;
- d) ||.

35. Результат выполнения следующего фрагмента кода: `!((1 || 0) && 0)`

- a) 0;
- b) 1;
- c) результат не может быть заранее определен.

36. После выполнения фрагмента кода

```
int a = 1, b = 2; if (a == b); cout << a << " = " << b << endl;
```

на экране появится:

- a) синтаксическая ошибка;
- b) `a = b`;
- c) вывод на экран не выполнится;
- d) `1 = 2`.

37. Логическая операция с большим приоритетом выполнения:

- a) ||;
- b) &&;
- c) !.

38. В блок-схеме для записи проверки условия используется фигура:

- a) эллипс;
- b) прямоугольник;
- c) параллелограмм;
- d) ромб.

39. Условный оператор относится к ___ алгоритму:

- a) к линейному;
- b) к циклическому;
- c) к разветвляющемуся;
- d) к основному.

40. Оператор `if else` позволяет определить действие:

- a) для истинного и ложного условий;
- b) только для истинного условия;
- c) только для ложного условия.

41. Синтаксическая ошибка допущена в условном операторе:

- a) `if (b == 0) cout << "Деление на ноль невозможно";`
- b) `if (a > b) max = a; else max = b;`
- c) `if (a > b) && (b > 0) c = a + b;`
- d) `if (a < b) min = a else min = b.`

42. Программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением, зарезервированному слову:

- a) other;
- b) all;
- c) void;
- d) default.

43. Общий формат оператора множественного выбора switch:

a) switch (N)

```
{  
  case M1:    S1;  
             break;  
  case M2:    S2;  
             break;  
  .....  
  default:    S;  
}
```

b) switch (N)

```
{  
  case M1:    S1;  
             break;  
  case M2:    S2;  
             break;  
  .....  
  else:       S;  
}
```

c) switch (N)

```
{  
  case M1:    S1;  
             * break;  
  case M2:    S2;  
             break;  
  .....  
  continue:  S;  
}
```

44. В результате выполнения следующей программы

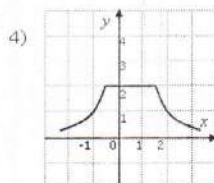
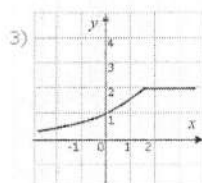
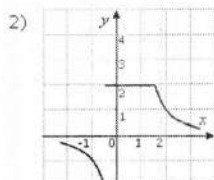
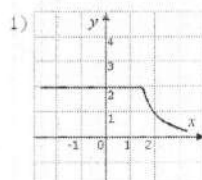
```
void main()  
{  
  int a, b, c, i;  
  b=9; c=4;  
  i=b/c;  
  switch (i)  
  { case 1: a=b+c;  
    case 2: { a=b-c; b=b+2; a=a+b-c;}  
    case 3: a=2*(b+c);  
    case 4: a=b*c;}  
  cout<<a;  
}
```

значение переменной a равно:

- a) 13;
- b) 12;
- c) 26;
- d) 36.

45. Выберите график функции, который соответствует следующему фрагменту программы:

```
if (x < -0.5) y = 1/abs(x);  
else if (x < 1.5) y = 2;  
else y = 1/(x-1);
```



46. Определите значение переменной c после выполнения следующего фрагмента программы

```
a = 30;  
b = 6;  
a = a / 2 * b;  
if (a > b)  
    c = a - 4 * b;  
else  
    c = a + 4 * b;
```

- a) 6;
- b) 66;
- c) 26;
- d) 54.

47. В результате работы фрагмента программы

```
k=2;  
d=5;  
if ((k+d)<3)  
{ m:=d-k;  
  cout<<"m="<<m;  
}  
else  
{ m:=d+k;  
  cout<<"m="<<m; }
```

на экране монитора появится число:

- a) 7;
- b) 3;
- c) 2;
- d) 5.

48. Укажите значение переменной y

x=3; n=2

switch (n)

```
{ case 1: y=x*x; break;  
  case 2: y=x*x*x; break;  
  case 3: y=x+3; break;  
}
```

a) 4;

b) 27;

c) 8.

49. Укажите бесконечные циклы:

a) for (i=1; i++;) printf("цикл\n");

b) for (i=0; i>=1; i++); printf("цикл\n");

c) for (i=10; i>6; i++); printf("цикл\n");

d) for (i=5; i>15; i++); printf("цикл\n").

50. В результате выполнения следующей программы:

void main()

```
{  
  int n,x;  
  n=1;  
  while (n<=5) {  
    x=n*n;  
    cout<<" "<<x;  
    n++;  
  }  
}
```

на экран будет выведено:

a) 1 4 9 16 25;

b) 1 2 3 4 5;

c) 5 4 3 2 1;

d) 25 16 9 4 1.

51. Найдите ошибочно записанные операторы:

a) while (true) i=i+1;

b) j=56;

i=1;

while (i<j div 7) i=i+1;

c) j=5; while (j>0) j=j+1;

d) j= -5; while (j<=0) j=j+1;

a) a;

b) a,c;

c) c;

d) a,d.

52. Операторы в теле цикла while (x < 100) будут выполняться до тех пор, пока:

a) x строго меньше 100;

b) x меньше или равен 100;

c) x равен 100;

d) x больше 100.

53. Для цикла while верно высказывание:

- a) тело цикла в любом случае выполнится хотя бы один раз;
- b) цикл с постусловием;
- c) цикл с предусловием.

54. Укажите соответствия:

- | | |
|-------------|-------------------------|
| a) for | 1) цикл с предусловием; |
| b) while | 2) цикл с постусловием; |
| c) do while | 3) цикл со счетчиком. |

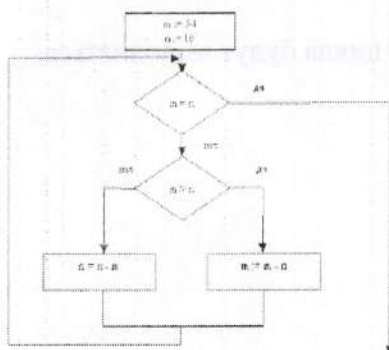
55. Хотя бы один раз обязательно выполнится:

- a) цикл с предусловием;
- b) цикл с постусловием;
- c) цикл со счетчиком.

56. Не является циклом в C++:

- a) do while;
- b) repeat until;
- c) for;
- d) while.

57. Значение переменной m после выполнения фрагмента алгоритма



будет равно ____.

58. Значение переменной S после выполнения операторов

```

i:=0; S:=0;
while (i<3)
{
    i:=i++;
    S:=S+i*i;
}
    
```

будет равно ____.

59. Дан фрагмент for (i=1, s=0; i<=3 ; i++). Один раз до цикла выполнится:

- a) i++;
- b) i<=3;
- c) i=1;
- d) s=0.

60. Дан фрагмент for (i=1, s=0; i<=5 ; i+=2) s+= i;

Укажите значение переменной s, после завершения цикла:

- a) 15;

- b) 4;
c) 9.

61. В результате выполнения программы

```
void main()
```

```
{
  int n, s;
  n = 30;
  s = 25;
  while (s < 425)
  {
    s = s + 40;
    n = n + 10;
  }
```

```
printf("%d", n);
}
```

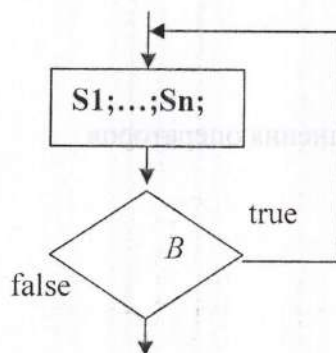
будет напечатано:

- a)
425;
b) 120;
c) 130;
d) 440.

62. Дан фрагмент `for (i=1, s=0; i<=5 ; i+=2) s+=i;` На каждом шаге цикла будут выполняться:

- a) `i+=2;`
b) `i<=5;`
c) `i=1;`
d) `s+=i;`
e) `s=0.`

63. Данная блок-схема



соответствует циклу:

- a) со счетчиком;
b) с предусловием;
c) с постусловием.

64. Если говорят, что программа "зациклилась", то это значит, что тело цикла:

- a) выполняется бесконечно долго;
b) не выполняется;
c) выполняется два раза;
d) выполняется только один раз.

65. Назначение циклической структуры:

- a) повторение идущих подряд одинаковых команд некоторое число раз;

- b) проверка условия в тексте;
- c) печать текста;
- d) повторение одной команды не более 10 раз.

66. Прервать выполнение операторов do, for, while и switch можно оператором:

- a) break;
- b) continue;
- c) exit.

67. Оператор Continue выполняет:

- a) переход к началу новой итерации цикла;
- b) переход к оператору, следующему за циклом;
- c) выход из цикла.

68. Прервать программу позволяет оператор:

- a) break;
- b) continue;
- c) exit.

ДЕ 4. Модульное программирование – 12

69. Подпрограмма – это:

- a) повторяющаяся группа операторов, оформленная в виде самостоятельной программной единицы. Она записывается однократно, а в соответствующих местах программы обеспечивается лишь обращением к ней по имени;
- b) независимая от основной программы группа операторов, оформленная в виде самостоятельной программной единицы. Она записывается однократно, обращение к ней из основной программы не происходит;
- c) повторяющаяся группа операторов, оформленная в виде самостоятельной программной единицы и записанная в отдельный файл.

70. Оператор, возвращающий значение функции:

- a) void;
- b) function;
- c) return;
- d) default.

71. Укажите правильный вызов функции, если функция была объявлена ранее.

- a) funct();
- b) funct x, y;
- c) int funct();
- d) funct.

72. Укажите тип возвращаемого значения функции `int func(char x, float v, double t);`

- a) float;
- b) double;
- c) int;
- d) char.

73. Выберите правильное (полное) определение функции.

- a) `void funct(int)`
`{`
`cout << "Hello"`
`}`

```
b) void funct(x)
{
cout << "Hello"
}
```

```
c) int funct();
```

```
d) int funct(int x)
{
return x = x + 1;
}
```

74. Выполнение программы на языке Си начинается с функции:

- a) main;
- b) void;
- c) default;
- d) include.

75. Максимальное число аргументов, которое можно передавать функции:

- a) 8;
- b) 256;
- c) 512;
- d) 1024.

76. После выполнения программы

```
#include <iostream.h>
#include <conio.h>
double Max(double a,double b)
{
    if (a>b)
        return a;
    else
        return b;
}
void main (void)
{
    double x,y,max;
    x=-5.126;
    y=-5.125;
    cout<<"max="<<Max(x,y)<<endl;
    getch();
}
```

на экран будет выведено:

- a) -5.126;
- b) -5.125;
- c) 0;

77. Установите соответствия:

- a) Аргумент
- b) Параметр

- 1) значение, передаваемое функции при вызове;
- 2) определяемая функцией переменная, которая принимает передаваемый функции аргумент.

78. Локальными называются переменные:

- a) которые объявлены в описании основной части, и их могут использовать любые функции данной программы;
- b) описанные внутри подпрограммы и используемые только внутри данной подпрограммы;
- c) описанные внутри подпрограммы и используемые как внутри данной подпрограммы, так и в основной программе.

79. Возвращаемое функцией по умолчанию значение принимает тип:

- a) void;
- b) int;
- c) float;
- d) bool.

80. В описании функции, которая не возвращает никаких результатов, используется тип:

- a) void;
- b) int;
- c) float;
- d) bool.

Ключи к тестовым заданиям

ДЕ1	
1	d
2	b
3	a
4	b
5	b
6	b
7	b
8	b
9	a
10	b,d
11	a
12	a
13	a, b, c
14	a,b

ДЕ2	
15	a
16	a,b,c
17	b
18	a
19	b
20	a2, b4, c1, d3
21	c
22	a
23	b
24	b
25	a
26	b

27	a,b
28	a
29	b
30	c
31	c
32	a

ДЕ3	
33	a
34	a
35	b
36	a
37	c
38	d
39	c
40	a
41	d
42	d
43	a
44	b
45	4
46	b
47	a
48	b
49	b
50	a
51	b
52	a
53	c

54	a3, b1, c2
55	b
56	b
57	4
58	14
59	c,d
60	c
61	c
62	d
63	c
64	a
65	a
66	a
67	a
68	c

ДЕ4	
69	a
70	c
71	a
72	c
73	d
74	a
75	b
76	b
77	a1, b2
78	b
76	b
80	a

При тестировании все верные ответы берутся за 100%, тогда оценка выставляется в соответствии с таблицей:

Процент выполнения задания	Оценка
85% и более	5 (отлично)
70-84%	4 (хорошо)
50-69%	3 (удовлетворительно)
менее 50%	2 (неудовлетворительно)

Задания для самостоятельной работы по теме «Оператор ветвления switch»

1. Написать программу, которая по номеру дня недели (целому числу от 1 до 7) выдает в качестве результата количество пар и их названия в вашей группе в этот день.
2. Написать программу, позволяющую по последней цифре числа определить последнюю цифру его квадрата.
3. Написать программу, которая по заданному году и номеру месяца m определяет количество дней в этом месяце.
4. Для каждой введенной цифры (0-9) вывести соответствующее ей название на английском и русском языке.
5. Написать программу, которая по номеру месяца выдает название следующего за ним месяца. (При $m=1$ получаем- февраль, 4- май и т.д.)
6. Напишите программу, которая анализирует человека по возрасту и относит к одной из четырех групп: дошкольник, ученик, работник, пенсионер. Возраст вводится с клавиатуры.
7. В старояпонском календаре был принят 12-летний цикл.
Годы внутри цикла носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. Написать программу, которая вводит номер некоторого года и печатает его название по старояпонскому календарю.
8. Написать программу, которая по введенному номеру времени года (1-зима, 2-весна, 3-лето, 4-осень) выдавала соответствующее этому времени года месяцы и количество дней в каждом месяце.
9. Написать программу, которая по номеру месяца (целому числу от 1 до 12) выдает в качестве результата названия сезона, соответствующее ему. Например: 5-весна.

10. Составьте программу, которая для целого числа k (от 1 до 99), введенного вами, напечатает фразу «Мне k лет», при этом в нужных случаях слово «лет» заменяя на слово «год» или «года».

Вопросы к экзамену (1 семестр)

1. Программирование на языках высокого уровня (на примере Си/C++). Язык Си: история, первоначальная область применения (системное программирование).
2. Принцип построения: компилируемые конструкции и интерпретируемые средства (библиотека стандартных функций). Раздельная трансляция, компилятор и редактор связей.
3. Ввод информации. Управляющие коды. Отображение специальных символов на экране монитора.
4. Форматированный вывод. Организация ввода данных, диалога. Комментарии.
5. Алфавит языка. Способы описания синтаксиса языка. Определение понятия «идентификатор». Служебные слова. Комментарии.
6. Типы данных. Имена и объявления. Описание данных, литералы.
7. Назначение стандартных заголовочных файлов. Компонировка программы.
8. Целые типы данных: `int`, `short`, `long`, `char`. Кодовый формат. Тип целых констант.
9. Арифметические операции для целых операндов.
10. Понятие преобразования данных. Явное преобразование типа. Правила преобразования операндов в процессе вычислений.
11. Плавающие типы данных: `float`, `double`. Кодовый формат. Описание данных, литералы. Тип констант с плавающей точкой.
12. Арифметические операции для вещественных операндов.
13. Математические функции стандартной библиотеки Си.
14. Оператор-выражение. Операции отношения. Результат вычисления отношений. Представление булевских значений «ложь», «истина» в Си.
15. Логические операции: `!`, `||`, `&&`. Операции простого и составного присваивания. Приоритеты операций.
16. Структурное программирование. Оператор ветвления `if`, синтаксис, выполнение оператора. Операции отношения.
17. Оператор ветвления `switch`, синтаксис, выполнение оператора.
18. Составной оператор. Сложные условия и логические операции: `!`, `||`, `&&`. Вложение операторов `if`, оператор `if ... else if ... else`. Оператор `switch`, выполнение, использование оператора `break`.
19. Структурное программирование. Операторы цикла с предусловием (`while`). Примеры применения циклов.
20. Изменение хода выполнения цикла с помощью операторов `break` и `continue`.
21. Операторы цикла с постусловием (`do-while`). Примеры применения циклов. Изменение хода выполнения цикла с помощью операторов `break` и `continue`.
22. Модульное программирование. Функции. Рациональные размер и количество параметров функции. Пример функции.
23. Функции. Аргументы и параметры. Передача аргументов по значению и по ссылке. Прототипы функций. Преобразование аргументов в точке вызова. Оператор `return`.

Вопросы к экзамену (2 семестр)

1. Рекурсивные подпрограммы. Понятие о рекурсии.
2. Сравнение рекуррентности и рекурсии. Прямой и обратный ход рекурсии. Точка останова работы рекурсивного алгоритма. Глубина рекурсии.
3. Одномерные массивы, пример использования. Связь массивов и указателей. Обращение к элементу массива, адресная арифметика.

4. Структурное программирование. Строки, кодовый формат, строковые литералы. Двумерные массивы. Функция индексации для двумерного массива.
5. Структуры, описание, пример использования. Операция выбора элемента структуры. Операции над структурой в целом.
6. Структуры как аргументы функций. Объявление типа: typedef. Размещение структур в памяти.
7. Модульное программирование. Функции. Рациональные размер и количество параметров функции. Пример функции. Аргументы и параметры.
8. Передача аргументов по значению и по ссылке. Прототипы функций. Преобразование аргументов в точке вызова. Оператор return.
9. Ввод и вывод (передача) с преобразованиями. Понятие набора данных и файла. Открытие и закрытие потоков. Функции передачи: fscanf и fprintf.
10. Управляющая строка, форматы при вводе и при выводе. Передача без преобразований (в кодовых форматах).
11. Время жизни и способ размещения данных. Спецификация класса памяти.
12. Статический способ размещения. Спецификаторы класса памяти extern и static.
13. Динамический процесс исполнения программы и концепция памяти auto / register. Данные с управляемым способом размещения, операции new и delete. Инициализация данных.
14. Объявления и определения. Область действия описаний. Структура программы на языке Си.
15. Переобъявления во вложенных блоках. Определения и объявления на внешнем уровне. Определения и объявления на внутреннем уровне.
16. Указатели, адресная арифметика, указатели и массивы. Введение в обработку списков.
17. Создание и уничтожение узлов списка, вставка и исключение узлов списка.
18. Препроцессор Си. Директива препроцессора #define.
19. Терминология: макроопределение, макрообращение, макрорасширение.
20. Макросы с параметрами, аналогия с функциями. Директива препроцессора #include. Директивы условной компиляции.

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Основная литература

1. Гросс Кр. С++ и платформа .NET 3.5 Framework: базовое руководство. 2-е изд. – М.: ООО "И. Д. Вильямс", 2009. – 480 с.
2. Нэш Т. С++ 2008. Ускоренный курс для профессионалов. – М.: ООО Издательский дом "Вильямс", 2008. – 576 с.
3. Подбельский В.В. Язык С. Базовый курс. – М.: Финансы и статистика, 2011. – 384 с.
4. Троелсен Э. Язык программирования С++ и платформа .NET 3.5. – М.: Вильямс, 2011. – 1344 с.
5. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений С++. – М.: КУДИЦ-ОБРАЗ, 2003. – 512 с.
6. Шилдт Г. С++ 4.0. Полное руководство. – М.: Издательский дом "Вильямс", 2011. – 1056 с.

8.2. Дополнительная литература

1. Нейгел К. и др. С++ 4.0 и платформа .NET 4.0 для профессионалов. – М.: ООО «И.Д. Вильямс», 2011. – 1440 с.

2. Пауэрс Л., Снелл М. Microsoft Visual Studio 2008. – СПб.: БХВ-Петербург, 2009. – 1200 с.
3. Петцольд Ч. Программирование для Microsoft Windows на C++. В 2-х томах. – М.: Издательско-торговый дом «Русская Редакция». – 2002. – Том 1. – 624 с.
4. Петцольд Ч. Программирование для Microsoft Windows на C++. В 2-х томах. – М.: Издательско-торговый дом «Русская Редакция». – 2002. – Том 2 – 576 с.
5. Рихтер Дж. CLR via C++. Программирование на платформе Microsoft .NET Framework 2.0 на языке C++. Мастер класс. 2-е изд. исправ. – СПб.: Питер, 2008. – 656 с.

8.3. Программное обеспечение и Интернет-ресурсы

Программное обеспечение:

1. Пакет Microsoft Office – офисное приложение.
2. Браузеры Интернет.
3. Инструментальная среда разработки Borland Turbo C (Dev-C++).

Интернет-ресурсы:

1. Образовательный портал [Электронный ресурс]. – Режим доступа: <http://kufas.ru/index.htm>.
2. Dev-C++ оболочка для программирования в Windowshttp [Электронный ресурс]. – Режим доступа: kpolyakov.spb.ru.

8.4. Методические указания и материалы по видам занятий

Методические указания предоставляются студентам в виде теоретических предпосылок (в электронном и печатном виде) к практическим и лабораторным работам.

9. Материально-техническое обеспечение дисциплины

Для осуществления образовательного процесса по дисциплине «Основы программирования» необходима компьютерная аудитория, а также лекционная аудитория, оборудованная мультимедийными средствами для проведения лекций-визуализаций. Для проведения практических и лабораторных занятий используется компьютерная аудитория, оснащенная 10 компьютерами, объединенными локальной сетью, с доступом в Интернет.

Методические пособия, используемые на лекционных, практических и лабораторных занятиях:

1. Программирование на языке высокого уровня Си. Часть I: практикум / сост.: О.В. Шестопал, О.В. Сташкова. – Тирасполь, 2009. – 65 с.
2. Программирование на языке высокого уровня СИ. Часть II: практикум. / сост.: О.В. Шестопал, О.В. Сташкова. – Рыбница, 2010. – 100 с.

10. Методические рекомендации по организации изучения дисциплины

Рабочая учебная программа по дисциплине «Основы программирования» составлена в соответствии с требованиями Федерального Государственного образовательного стандарта ВО по направлению 09.03.04 «Программная инженерия» и учебного плана по профилю подготовки «Разработка программно-инструментальных систем»

Изучение дисциплины «Основы программирования» включает лекционные, практические и лабораторные занятия.

Во время выполнения заданий лабораторной работы в учебной аудитории студент может консультироваться с преподавателем, определять наиболее эффективные методы решения поставленных задач. Если какая-то часть задания остается не выполненной, студент может продолжить её выполнение во время внеаудиторной самостоятельной работы. При подготовке к лабораторным занятиям студентам необходимо самостоятельно изучить рекомендуемую литературу, ответить на контрольные вопросы.

Для выполнения лабораторной работы необходимо:

1. Изучить краткие теоретические сведения, необходимые для успешного выполнения конкретной работы.
2. Внимательно изучить все примеры заданий, рассмотренные в лекции и представленные в описании лабораторной работы.
3. Ответить на контрольные вопросы, предложенные в лабораторной работе.
4. Выполнить индивидуальные задания.
5. Оформить отчет о выполненной лабораторной работе.

Отчет оформляется в тетради или распечатывается на листах формата А4 и представляется преподавателю на проверку по завершению изучения темы. Отчет должен содержать: название темы, цель работы, ход работы, вывод. Отчет о лабораторной работе принимает преподаватель во время лабораторного занятия. В процессе защиты оценивается самостоятельность работы, понимание механизма работы среды, умение анализировать результаты выполнения работы.

Работа с информационными источниками считается одним из основных видов самостоятельной работы. Для приобщения обучаемых к поиску, к исследовательской работе, для развития их творческого потенциала следует по возможности избегать прямого руководства работой обучающихся при выполнении ими тех или иных заданий, чаще выступать в роли консультанта, эксперта, коллеги-исследователя.

Текущий контроль усвоения знаний по дисциплине предполагает использование разных форм контроля, в том числе проверки выполнения индивидуальных и лабораторных заданий. Итоговый контроль может осуществляться в форме экзамена, теста. Вопросы к экзаменам и образец тестовых заданий приведены. Выполнение индивидуальных заданий на практических занятиях и лабораторных заданий является необходимым условием для допуска к экзамену.

11. Технологическая карта дисциплины

Курс: I группа: РФ17ДР62ПИ семестр: 1, 2

Преподаватели-лекторы: Чебан Наталья Сергеевна, Гарбузняк Елена Сергеевна

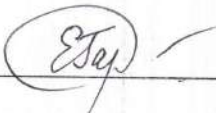
Преподаватели, ведущие практические и лабораторные занятия: Чебан Н.С., Гарбузняк Е.С.

Кафедра информатики и программной инженерии

Наименование дисциплины / курса	Уровень/ступень образования (бакалавриат, специалитет, магистратура)	Статус дисциплины в рабочем учебном плане (А, Б, В, Г)	Количество зачетных единиц / кредитов	
Основы программирования	бакалавриат	Б	12	
Смежные дисциплины по учебному плану:				
«Иностранный язык», «Теоретическая информатика», «Введение в алгоритмы», «Типы и структуры данных», «Объектно-ориентированное программирование»				
ВВОДНЫЙ МОДУЛЬ				
(входной рейтинг-контроль, проверка «остаточных» знаний по смежным дисциплинам)				
Тема, задание или мероприятие входного контроля	Виды текущей аттестации	Аудиторная или внеаудиторная	Минимальное количество баллов	Максимальное количество баллов

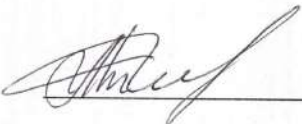
Итого:				
БАЗОВЫЙ МОДУЛЬ (проверка знаний и умений по дисциплине)				
Тема, задание или мероприятие текущего контроля	Виды текущей аттестации	Аудиторная или внеаудиторная	Минимальное количество баллов	Максимальное количество баллов
Итого:				
ДОПОЛНИТЕЛЬНЫЙ МОДУЛЬ				
Тема, задание или мероприятие дополнительного контроля	Виды текущей аттестации	Аудиторная или внеаудиторная	Минимальное количество баллов	Максимальное количество баллов
Итого максимум:				

Составители:  О.В. Сташкова, ст. преподаватель

 Е.С. Гарбузняк, ст. преподаватель

 Н.С. Чебан, преподаватель

Зав. кафедрой информатики
и программной инженерии

 Л.А. Тягульская, доцент