

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САХАЛИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
для студентов по выполнению
практических занятий
по дисциплине
**ОП.11. ОСНОВЫ АЛГОРИТМИЗАЦИИ И
ПРОГРАММИРОВАНИЯ**
(часть 3)

09.00.00 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
по специальности
09.02.05 Прикладная информатика (по отраслям)
(базовый уровень подготовки)

Квалификация: техник-программист
Форма обучения: очная

Южно-Сахалинск
2014

УТВЕРЖДАЮ
Зам. директора по УР
К.М. Ким
« 26 » сентября 20 19 г.

Разработчик(и): Вилькер Е.Ю., преподаватель

Одобрено на заседании ПЦК

информационных
технологий

Протокол № 1 от «23» сентября 2019 г.

Председатель ПЦК

Вилькер Е.Ю. Вилькер Е.Ю.

Согласовано

Вилькер Е.Ю.
подпись

Вилькер Е.Ю.
Ф.И.О.

зав.отделением

информационных

СОДЕРЖАНИЕ

№		Страница
1	ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	4
2	ТЕМАТИЧЕСКИЙ ПЛАН	6
3	ПЕРЕЧЕНЬ ЗАДАНИЙ К ПРАКТИЧЕСКОЙ АУДИТОРНОЙ РАБОТЕ СТУДЕНТОВ	
	№ 1 4.1.3. Обработка одномерных массивов.	9
	№ 2 5.3.2. Создание приложения.	12
	№ 3 5.4.3. Объявление класса, создание экземпляров класса.	14
	№ 4 5.4.4. Создание наследованного класса. Перегрузка методов.	17
	№ 5 5.5.3. Создание проекта с использованием кнопочных компонентов и компонентов для работы с текстом.	21
	№ 6 5.5.4. Создание проекта с использованием кнопочных компонентов ввода и отображения чисел, дат, времени и компонентов стандартных диалогов и системы меню.	25
	№ 7 5.6.4. Разработка оконного приложения с одной или несколькими формами.	30
4	ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ	35

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания по выполнению практических работ по дисциплине «Основы алгоритмизации и программирования» предназначены для реализации государственных требований к минимуму содержания и уровню подготовки выпускников по специальности 230701 «Прикладная информатика (в экономике)» среднего профессионального образования.

В результате изучения дисциплины и выполнения практических работ студент *должен уметь:*

- Работать в среде программирования;
- Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

знать:

- Этапы решения задачи на компьютере;
- Типы данных;
- Базовые конструкции изучаемых языков программирования;
- Принципы структурного и модульного программирования;
- Принципы объектно-ориентированного программирования.

Формируемые компетенции:

Общие:

- ОК 1. Понимать сущность и социальную значимость своей будущей специальности, проявлять к ней устойчивый интерес;
- ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество;
- ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность;
- ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития;
- ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности;
- ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями;
- ОК 7. Брать на себя ответственность за работу членов команды, за результат выполнения заданий;
- ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации;
- ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности;
- ОК 10. Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей);

Профессиональные:

- ПК 2.1. Осуществлять сбор и анализ информации для определения потребностей клиента;
- ПК 2.2. Разрабатывать и публиковать программное обеспечение и информационные ресурсы отраслевой направленности со статическим и динамическим контентом на основе готовых спецификаций и стандартов;
- ПК 2.3. Проводить отладку и тестирование программного обеспечения отраслевой направленности;
- ПК 2.4. Проводить адаптацию отраслевого программного обеспечения;

- ПК 2.5. Разрабатывать и вести проектную и техническую документацию;
- ПК 2.6. Участвовать в измерении и контроле качества продуктов;
- ПК 3.1. Разрешать проблемы совместимости программного обеспечения отраслевой направленности;
- ПК 3.2. Осуществлять продвижение и презентацию программного обеспечения отраслевой направленности;
- ПК 3.3. Проводить обслуживание, тестовые проверки, настройку программного обеспечения отраслевой направленности;
- ПК 4.1. Обеспечивать содержание проектных операций;
- ПК 4.4. Определять ресурсы проектных операций;

Важным направлением профессиональной подготовки специалиста в области программирования в компьютерных системах является формирование умений решать задачи с использованием компьютера и его программного обеспечения.

Структура учебной работы студентов в рамках данного практикума предусматривает практические занятия в компьютерном классе, в ходе которых под руководством преподавателя осуществляется разработка и реализация на компьютере конкретных программ в соответствии с предложенной тематикой. При этом в ходе самостоятельной индивидуальной работы студент должен осуществить полный цикл разработки программ, включая подготовку отчета по каждой из лабораторных работ.

Настоящие методические указания не предусматривают следование какому-либо конкретному языку программирования для использования при решении задач, однако на практике используется язык Turbo Pascal. Это связано со следующим: во-первых, данный язык является очень распространенным, обладает мощными средствами для создания эффективных программ; набор операторов и управляющих конструкций данного языка аналогичен конструкциям, имеющимся в других языках программирования (C, Ada и др.). Во-вторых, этот язык является основой языка Object Pascal, на котором, в свою очередь, базируется система программирования Delphi. При этом следует иметь в виду, что содержание дисциплины «Основы программирования», изучение которой рекомендуется осуществлять во взаимосвязи с выполнением заданий данного практикума, отражает именно современный объектно-ориентированный подход к программированию, который на практике может быть реализован с использованием системы программирования Turbo Pascal.

Умения в области программирования, сформированные в процессе освоения «Практикума по основам программирования», составят базу, на основе которой в дальнейшем могут быть эффективно решены задачи практической подготовки. У студентов в ходе изучения соответствующего учебного материала должны быть сформированы знания и практические умения, достаточные для самостоятельного составления типовых алгоритмов и программ.

На практические работы по дисциплине «Основы алгоритмизации и программирования» отводится 44 часа (тематический план прилагается).

Методические указания включают в себя:

- Темы;
- Количество часов на изучение темы;
- Требования к содержанию и объему выполняемых заданий;
- Задания;
- Формы контроля;
- Примеры выполнения заданий и решения задач;
- Критерии оценок.

ТЕМАТИЧЕСКИЙ ПЛАН:

Разделы, темы	Аудиторные часы	
	теория	Практи ч. з.
3 семестр – 32 часа (18 т. + 14 пр.)		
Раздел 1. Введение в программирование	6	4
1.1. Основы алгоритмизации.		
1.1.1. Алгоритмы. Свойства и способы описания алгоритмов. Основные конструкции алгоритмического языка: линейный, ветвление, цикл.	2	
1.1.2. Составление алгоритмов линейной и разветвляющейся структуры		2
1.1.3. Составление алгоритмов циклической структуры		2
1.2. Языки программирования.		
1.2.1. Развитие языков программирования. Обзор ЯП. Области применения ЯП. Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере.	2	
1.3. Типы данных.		
1.3.1. Переменные и константы. Объявление объектов данных. Типы данных. Простые и производные типы данных. Структурированные типы данных.	2	
Раздел 2.	6	4
Операторы языка программирования		
2.1. Операторы языка программирования.		
2.1.1. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных.	2	
2.1.2. Оператор присваивания. Составной оператор. Условный оператор. Оператор выбора. Цикл.	2	
2.1.3. Составление программ линейной и разветвляющейся структуры структуры.		2
2.1.4. Составление программ циклической структуры.		2
Контрольная работа. Составление программ линейной, разветвляющейся и циклической структуры.	2	
Раздел 3.	6	6
Структурное и модульное программирование		
3.1. Процедуры и функции.		
3.1.1. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Механизм передачи параметров. Рекурсия.	2	
3.1.2. Организация процедур и функций. Использование процедур и функций.		2
3.1.3. Применение рекурсивных функций.		2
3.2. Структуризация в программировании. Основы структурного программирования. Методы структурного программирования.	2	
3.3. Модульное программирование.		
3.3.1. Модульное программирование. Понятие модуля. Структура модуля. Стандартные модули. Компиляция и компоновка программы.	2	
3.3.2. Программирование модуля. Создание библиотеки подпрограмм.		2
4 семестр – 34 часа (18 т. + 16 пр.)		
Раздел 4. Структуры данных	18	16
4.1. Массивы.		
4.1.1. Объявление массива. Инициализация. Действия над массивами.	2	

Заполнение массива данными. Вывод элементов массива.		
4.1.2. Обработка массива. Удаление и вставка элементов в массив.	2	
4.1.3. Обработка одномерных массивов.		2
4.1.4. обработка двумерных массивов.		2
4.2. Строки.		
4.2.1. Символьный и строковый типы. Объявление типов. Поиск, удаление, замена и добавление символов в строке. Операции со строками. Стандартные функции и процедуры для работы со строками.	2	
4.2.2. Работа со строковыми переменными. Использование стандартных функций и процедур для работы со строками.		2
4.3. Множества.		
4.3.1. Понятие множества. Объявление множества. Операции над множествами.	2	
4.3.2. Работа с данными типа множество.		2
4.4. Записи.		
4.4.1. Определение типа запись. Правила работы с записями.	2	
4.4.2. Составление программ, содержащих записи.		2
4.5. Файлы.		
4.5.1. Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие, закрытие, запись в файл и чтение из файла последовательного доступа.	2	
4.5.2. Файлы произвольного доступа. Порядок работы с файлами произвольного доступа. Создание структуры записи. Открытие, закрытие, запись и считывание из файла произвольного доступа. Стандартные процедуры и функции для файлов разного типа.	2	
4.5.3. Работа с файлом последовательного доступа.		2
4.5.4. Работа с файлом произвольного доступа.		2
4.6. Указатели.		
4.6.1. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных. Структуры данных на основе указателей.	2	
4.6.2. Использование указателей для организации связанных списков.		2
<i>Решение задач по теме: «Структуры данных на основе указателей»</i>		
Контрольная работа. Структуры данных.	2	
5 семестр – 30 часов (16 т. + 14 пр.)		
Раздел 5.		
Объектно-ориентированное программирование	16	14
5.1. Основные принципы объектно-ориентированного программирования (ООП).		
История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиформизм. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства	2	
5.2. Интегрированная среда разработчика.		
5.2.1. Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта.	2	

Выполнение проекта. Настройка среды и параметров проекта.		
5.2.2. Изучение интегрированной среды разработчика.		2
5.3. Этапы разработки приложения.		
5.3.1. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Программирование приложения. Тестирование, отладка приложения. Создание документации.	2	
5.3.2. Создание приложения.		2
5.4. Иерархия классов.		
5.4.1. Классы объектно-ориентированного языка программирования: виды, назначение, свойства, методы, события. Объявление класса, свойств и методов экземпляра класса.	2	
5.4.2. Наследование. Перегрузка методов.	2	
5.4.3. Объявление класса, создание экземпляров класса.		2
5.4.4. Создание наследованного класса. Перегрузка методов.		2
5.5. Визуальное событийно-управляемое программирование.		
5.5.1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления. Свойства компонентов (элементов управления). Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства.	2	
5.5.2. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Процедуры, определенные пользователем: синтаксис, передача аргументов. Вызов событий.	2	
5.5.3. Создание проекта с использованием кнопочных компонентов и компонентов для работы с текстом.		2
5.5.4. Создание проекта с использованием кнопочных компонентов ввода и отображения чисел, дат, времени и компонентов стандартных диалогов и системы меню.		2
<i>Выполнение индивидуального проектного задания по теме «Объектно-ориентированное программирование»</i>		
5.6. Разработка оконного приложения.		
5.6.1. Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Создание процедур обработки событий. Компиляция и запуск приложения.	2	
5.6.4. Разработка оконного приложения с одной или несколькими формами.		2
ВСЕГО:	52	44