ТИПОВАЯ ПРОГРАММА

по дисциплине «Алгоритмическое языки и программирования»

для направления подготовки 531200 «Компьютерная лингвистика»,
Академическая степень: бакалавр

Составитель: **Туракеева М. У.** – ст. преп. кафедры Информационных технологий факультета Лингвистики

СТРУКТУРА ПРОГРАММЫ

II. Организационно-методический раздел
I.1. Цели и задачи дисциплины
I.2. Место дисциплины в структуре ООП высшего профессионального образования
I.3. Требования к результатом освоения содержания дисциплины
I.4. Объем дисциплины и виды учебной работы
I.5. Формы контроля
I.6. Технологическая карта дисциплины
II. Содержание программы учебной дисциплины
II.1 Содержание разделов дисциплины для 3-курса
II.2 Вопросы и задания для самостоятельной работы студентов
III. Оценочные средства по итогам освоения дисциплины
III.1 Примерные темы проектов
III.2 Примерные вопросы к экзамену
IV Образовательные технологии
IV.1 Учебно-методические: основная и дополнительная литература
IV.2 Информационное и материально-техническое обеспечение дисциплины: база-данных, информационно-справочные и поисковые системы
IV.3. Материально-техническое обеспечение дисциплины

I. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

I.1. Целью освоения дисциплины «Алгоритмическое языки и программирования» является:

Дисциплина "Алгоритмические языки и программирование" является обучение студентов навыкам проектирования программ с использованием современных технологий структурного программирования, освоение и получение навыков работы на современных вычислительных средствах, развитие умения проведения отладки и тестирования программ.

Основными задачами преподавания данной дисциплины являются изучение студентами теоретических основ алгоритмизации и проектирования программ, процесса структурного программирования, его технологий, приемов программирования на конкретных алгоритмических языках, основ организации вычислительного процесса на ЭВМ.

Цель дисциплины:

формирование базовых профессиональных компетенций по разработке программного обеспечения на языке программирования высокого уровня.

Задачи, решение которых обеспечивает достижение цели:

- знание современных технологий программирования (структурное, модульное программирование);
- освоение принципов проектирования алгоритмов задач;
- знание технологии разработки алгоритмов и программ;
- знание методов отладки и тестирования программ;
- знание методов решения задач на ЭВМ в различных режимах;
- умение ставить задачу и разрабатывать алгоритм ее решения;
- умение разрабатывать основные программные документы;
- умение использовать прикладные системы программирования;
- владение навыками разработки и отладки программ на одном из алгоритмических процедурных языков программирования высокого уровня;
- формирование научного мировоззрения будущего специалиста.

Задачи дисциплины:

1. Овладение приемами работы с современными пакетами прикладных программ, обеспечивающих широкие возможности обработки информации;

2. Формирование у студентов представления о возможностях использования средств вычислительной техники, технологий программи-

рования при решении различного вида экономических, производственных и учебных задач;

3. Обучение принципам построения информационных моделей и проведения анализа полученных результатов;

4. Развитие навыков алгоритмического мышления, получение знаний и практических навыков по алгоритмизации, разработке, отладке и тестированию программ;

5. Овладение эффективными приемами работы в современных средах программирования;

Пререквизиты – это дисциплины, содержащие знания, умения и навыки, необходимые для освоения изучаемой дисциплины, соответственно до изучения данной дисциплины определяют ее преемственность. Пререквизитами данного курса являются знания студентов по информатике.

Постреквизиты– это дисциплины, для изучения которых требуются знания, умения и навыки, приобретаемые по завершении 1-го курса, т. е. это обязательные для освоения дисциплины, соответственно после изучения данного предмета.

I.2. Место дисциплины в структуре ООП бакалавриата

Дисциплина «Алгоритмические языки и программирование» относится к базовой части профессионального цикла дисциплин (Б. 3. 6. 1, Б. 3. 6. 2).

Для освоения дисциплины «Алгоритмические языки и программирование» студенты используют знания, умения и виды деятельности, формируемые одновременно с изучением дисциплин «Физика», «Математическая логика и теория алгоритмов» математического и естественнонаучного цикла дисциплин». Компетенции, формируемые при изучении дисциплины, необходимы для самоорганизации работы студентов при выполнении ими курсовых работ, выпускной квалификационной работы.

Освоение дисциплины «Алгоритмические языки и программирование» является необходимой для последующего изучения дисциплин:

1. базовой части профессионального цикла:
   - «ЭВМ и периферийные устройства»;
   - «Защита информации»;
2. вариативной части профессионального цикла дисциплин:
   - «Теория языков программирования и методы трансляции»;
   - «Web-программирование»;
   - «Распознавание образов и обработка изображений»;

– «Основы теории управления»;
– «Модели и методы исследования операций»;

3. дисциплин по выбору профессионального цикла, а также для успешного прохождения учебной практики и итоговой государственной аттестации.

I.3. Требования к результатам освоения содержания дисциплины

В результате изучения данной дисциплины студент формирует и демонстрирует следующие общекультурные и профессиональные компетенции:

Общекультурные:

- владеет культурой мышления, способен к обобщению, анализу, восприятию информации, постановке цели и выбору путей её достижения (ОК - 1):
- умеет работать с информацией при разработке требований к программе: анализировать исходные данные задачи и результаты работы программы, определять функциональные требования к разрабатываемой программе, разрабатывать наборы тестовых данных;
- умеет анализировать методы решения задачи и обосновывать выбранный метод;
- умеет свести решение задачи к решению подзадач;
- демонстрирует культуру мышления при разработке алгоритмов подзадач и задачи в целом;

Профессиональные:

- обобщает данные при написании кода программы;
- оценивает степень соответствия разработанной программы требованиям, приведенным в спецификациях;
- демонстрирует культуру мышления при защите лабораторных работ, на практических занятиях и на экзамене;
- осознает социальную значимость своей будущей профессии, обладает высокой мотивацией к выполнению профессиональной деятельности путем выявления связей между конкретным знанием и его применением;

Проектно-технологическая деятельность: способность разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования (ПК - 12):

- знает основные критерии качества ПО;
- знает фазы жизненного цикла программного обеспечения;

- демонстрирует умение разработки спецификаций на разрабатываемую программу;
- демонстрирует умение разрабатывать программы на базе модульной и процедурно-ориентированной технологии;
- знает основные элементы и возможности языка программирования Си;
- знает и умеет применять основные алгоритмы обработки простых, структурированных и абстрактных данных на базе языка программирования Си;
- знает и умеет применять возможности интегрированной среды при написании программы, ее тестировании и отладке.
- дисциплин по выбору профессионального цикла, а также для успешного прохождения учебной практики и итоговой государственной аттестации.

Профессиональные компетенции:

- Способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-12);
- Готовность участвовать в работах по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем (ПК-15);
- Способность использовать технологии разработки объектов профессиональной деятельности в системах массовой информации, дизайна и медиаиндустрии, а также на предприятиях различного профиля и осуществлять все виды деятельности в условиях экономики информационного общества (ПК-18);
- Способность осуществлять организацию рабочих мест, их техническое оснащение, размещение компьютерного оборудования (ПК-19);
- Способность проводить сбор, анализ научно-технической информации, отечественного и зарубежного опыта по тематике исследования (ПК-23);
- Способность к инсталляции, отладке программных и настройке технических средства для ввода информационных систем в опытную эксплуатацию (ПК-29);
- Готовность адаптировать приложения к изменяющимся условиям функционирования (ПК-34);

В результате освоения дисциплины «Алгоритмические языки и программирование» студент должен:

Знать:

- ядро языка программирования, его синтаксис и семантику;
- основы проектирования программ;
- типовые алгоритмы работы с данными.

Уметь:

- читать литературу по информатике, рассчитанную на конечного пользователя (документацию к программным средствам, книги с описанием возможностей и технологиями использования ПО);
- использовать термины предметной области в устной речи;
- осознанно подходить к выбору среды разработки программного продукта, оценивая ее возможности, достоинства и недостатки;
- осваивать самостоятельно технологии использования незнакомых программных средств;
- анализировать содержание задач, создавать их формализованную модель;
- реализовывать на языке программирования высокого уровня готовые алгоритмы, модернизировать их и разрабатывать собственные в соответствии с условиями поставленной задачи.

Владеть:

- общими приемами разработки консольных и Windows-приложений; приемами работы в средах программирования (составление, отладка и тестирование программ; разработка и использование интерфейсных объектов)

I.4. Объем дисциплины и виды учебной работы

Виды учебной работы	Всего		Семестр	
	часов	кредиты	5	6
Аудиторные занятия (всего)	76	5	45	38
В том числе				
Лекционнные занятия	38		22	16
Практические занятия	**38**		**23**	**14**
Самостоятельная работа (всего)	**76**		**45**	**30**
Вид промежуточной аттестации			**аттестация**	**экзамен**
Общая трудоемкость часов	**150**	5		

I.5. Формы контроля: Для контроля усвоения данного курса учебным планом предусмотрен ***Экзамен.***

Курс направлен на приобретение и развитие следующих компетенций:

1. Основы языки программирования.

2. Методы работы с программированием.

Текущий контроль осуществляется лектором и преподавателем, ведущим практические занятия.

Контроль за освоением дисциплины осуществляется в каждом дисциплинарном разделе отдельно.

1.6. Технологическая карта дисциплины

Наименование дисциплины	Уровень/ ступень образования (бакалавриат/ магистратура)	Количество зачетных единиц/ кредитов	Форма отчетности	Курс, семестр
Алгоритмические языки и программирование	бакалавриат	5	Экзамен	3 курс 5-6 семестр

БАЗОВЫЙ МОДУЛЬ (проверка знаний и умений по дисциплине)				
Тема или задание текущей аттестационной работы	Виды текущей аттестации	Аудиторная или внеаудиторная	Минимальное количество баллов	Максимальное количество баллов
Выполнение заданий в течение учебного процесса	Своевременное выполнения домашних и проектных работ	Внеаудиторная или аудиторная	30	40
Выполнение заданий самостоятельную работу (Написание реферата, проектных работ по задан. теме)	Выполнение определенного задание по СРС	внеаудиторная	5	15
Знание базовых понятий и примеров по основным темам и умение решить практических задач	Контрольная работа, тестирование	Аудиторная и внеаудиторная	25	30
Итого:			60	85

ДОПОЛНИТЕЛЬНЫЙ МОДУЛЬ				
Тема или задание текущей аттестационной работы	**Виды текущей аттестации**	**Аудиторная и внеаудиторная**	**Минимальное количество баллов**	**Максимальное количество баллов**
Проект, презентация	Устный ответ	Аудиторная	4	10
Посещение занятий	Присутствие	Аудиторная	1	5
Итого:			5	15
Итого всего:			65	100

Все оценки складываются из результатов накопленных баллов по каждому модулю и для финального контроля выводится средний балл для окончательной оценки за курс:

Шкала перевода баллов в оценки следующее:

1 модуль -70 баллов	СРС-15 баллов
0-51 «неуд»	0-3 «неуд»
52-62 «удовл»	4-8 «удовл»
63-74 «хор»	9-11 «хор»
75-85 «отл»	12-15 «отл»

Проверка знаний: 3 контрольные работы (или тестирование) по основным темам и 2 проектные работы.

Дополнительные требования для студентов, отсутствующих на занятиях по уважительной причине: устное сквозное собеседование с преподавателем по проблеме пропущенных практических занятий, обязательное выполнение аудиторных проверочных и письменных работ.

Итоговая форма контроля – экзамен.

II. СОДЕРЖАНИЕ ПРОГРАММЫ УЧЕБНОГО КУРСА

II.1. Содержание разделов дисциплины

Тема 1. Основы программирования. Введение в программирование. Основные типы данных и алгоритмических конструкций

Тема 2. Технология разработки программ. Процедуры и функции.

Тема 3. Структурированные типы данных и основные алгоритмы работы с ними.

Тема 4. Файлы. Типы файлов. Основные алгоритмы работы с файлами.

Тема 5. Расширение возможностей языка программирования.

Тема 6. Типы данных, определяемые программистом. Классификация типов данных. Перечисляемые, интервальные, структурные типы данных.

Тема 7. Указатели. Динамические структуры данных. Интегрированные среды разработки программных продуктов. Объектная модель.

Тема 8. Технология создания программ в среде визуального программирования

Тема 9. Введение в визуальное программирование Введение в windows-формы. Интегрированные среды разработки программных продуктов. Объектная модель. Компоненты. Свойства и события.

Тема 10. Организация ввода и вывода информации. Основные события. Компоненты, с помощью которых можно организовать ввод и вывод информации. Чтение информации из файла.

Примерный план практических заданий

1. Этапы создания программного обеспечения.

2. Характеристика языка программирования Pascal (C/C++, C#). Описание и характеристика сред: состав среды, основные возможности. Состав и назначение отдельных файлов проекта.

3. Общее понятие о типах данных. Классификация типов данных. Машинное представление типов данных. Более подробно о простых типах данных: Порядковые (перечислимые – стандартные и интервальные) и вещественные.

4. Общее понятие о типах данных. Классификация типов данных. Структурные типы данных. Понятие о множестве. Операции со множествами. Примеры задач, для решения которых эффективно применять тип данных множество.

5. Общее понятие о типах данных. Классификация типов данных. Структурные типы данных. Записи. Записи с вариантами. Приемы работы с записями. Примеры задач, для решения которых эффективно применять тип данных запись.

6. Общее понятие о типах данных. Классификация типов данных. Структурные типы данных. Файлы. Типизированные и текстовые файлы. Приемы работы с файлами. Примеры обработки файлов.

7. Общее понятие о типах данных. Классификация типов данных. Структурные типы данных. Массивы. Одномерные и двумерные. Основные операции с массивами. Библиотечные модули для работы с массивами. Примеры задач, для решения которых эффективно применять тип данных массив.

8. Понятие компонента. Свойства и события. Событийное программирование. Компоненты, позволяющие реализовывать ввод и вы-

вод данных в Windows-приложениях: метки, поля ввода, поле редактора. Особенности использования.

9. Понятие компонента. Свойства и события. Событийное программирование. Компоненты, позволяющие реализовывать развилки в Windows-приложениях: радиокнопки, чекбоксы, листбоксы. Особенности использования.

10. Погрешности компьютерных вычислений. Причины возникновения. Варианты проявления. Исключительные ситуации. Варианты перехвата управления при исключительных ситуациях.

11. Понятие о процедурном программировании. Процедуры и функции: сходство и различие. По какому принципу выделяются процедуры и функции. Событийные процедуры в Windows-приложениях. Понятие о модуле. Понятие о библиотечной процедуре. Стандартные библиотеки. Библиотеки программиста.

12. Вычисление с заданной точностью. Класс задач, приводящих к алгоритмам нахождения сумм с заданной точностью. Описание алгоритма решения таких задач. Понятие о рекуррентных формулах. Преимущества использования рекуррентных записей.

II.2. Вопросы и задания для самостоятельной работы студентов

1. Принципы работы с массивами данных. Алгоритмы сортировки (не менее 3х), алгоритм двоичного поиска. Нахождение экстремальных элементов в массиве.
2. Принципы создания различных вариантов пользовательского интерфейса: меню, элементы управления и т. д.
3. Графические и мультимедийные возможности языка и среды программирования.
4. Понятие о компонентах. Свойства компонентов. События компонентов. Иерархия компонентов. Примеры использования компонентов, как объектов класса Object. Массив компонентов и действия над элементами массива.
5. Понятие о динамической памяти. Указатель. Действия над указателями. Работа со списками: создание первого элемента. Добавление очередного элемента. Удаление элемента. Общие правила работы с динамическими списками. (Схемы и фрагменты кода)
6. Понятие о дереве. Принцип формирования двоичного дерева. Стек и очередь. Сходство и различие. Основные процедуры обработки. (Схемы и фрагменты кода).
7. Общая характеристика среды программирования VisualStudio. Понятие о платформе Net. Современные требования к языкам программирования и средам разработки.

8. Сравнение синтаксиса двух языков программирования. Особенности работы с данными разных типов.
9. Общие принципы отладки и тестирования программ.
10. Принципы составления тестов для отладки. Основные типы ошибки, возникающие при разработке и меры по их предотвращению.

III. ОЦЕНОЧНЫЕ СРЕДСТВА ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

III.1. Примерные вопросы к экзамену

1. Понятие языков программирования.
2. Алгоритм, свойства алгоритма, способы описания алгоритма
3. Понятие программы, виды языков программирования
4. Машинный код.
5. Назначение трансляторов, компиляторов и интерпретаторов программ, их отличие.
6. Уровни и поколения языков программирования.
7. Языки программирования высокого уровня?
8. Какие языки программирования высокого уровня.
9. Понятие структурного программирования.
10. Понятие подпрограммы, структура подпрограмм.
11. Понятие процедуры и функции, их назначение
12. Понятие визуального программирования. Какие языки визуального программирования знаете?
13. Технология разработки программ, понятие постановки задачи

IV. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

IV.1. Учебно-методические: основная и дополнительная литература

Литература

1. Немнюгин С. А. TurboPascal. Программирование на языке высокого уровня: учебник для вузов. – СПб: Из-во «Питер», 2006 г. – 544 с.
2. Павловская Т. А. C#. Программирование на языке высокого уровня Учеб. для вузов 1 изд. - СПб.: Питер, 2009, 432 с.: ил.
3. Павловская Т. А. C/C++. Программирование на языке высокого уровня: учебное пособие для вузов. СПб.: Питер, 2006, 180 с.
4. Вирт Н. Алгоритмы и структуры данных. Изд. 3-е, - М: Мир, 1988, 358 с.
5. Давыдов В. Г. Программирование и основы алгоритмизации: учеб. пособие 2-е изд. - М.: Высшая школа, 2005, 447 с.: ил.

6. Емельянов В. И. Основы программирования на Delphi: учеб. пособие /Емельянов В. И., Воробьев В. И., Тюрина Т. П.,- М.: Высшая школа, 2005.

Дополнительная литература

1. Джоунз Г. Программирование на языке ОККАМ. - М.: Мир, 1989.
2. Сырков Б. Ю., Матвеев С. В. Программное обеспечение мультитранспьютерных систем. - М.: Диалог-МИФИ, 1992.
3. Хоар Ч. Взаимодействующие последовательные процессы. - М.: Мир, 1989.
4. Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J. Parallel Programming in OpenMP. - Morgan Kaufinann Publishers, 2000
5. Geist G. A., Beguelin A., Dongarra J., Jiang W., Manchek B., Sunderam V. PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Network Parallel Computing. MIT Press, 1994.
6. Group W, Lusk E, Skjellum A. Using MPI. Portable Parallel Programming with the Message-Passing Interface. - MIT Press, 1994. (htp: //www. mcs. anl. gov/mpi/index. html)
7. Kumar V., Grama A., Gupta A., Karypis G. Introduction to Parallel Computing. - The Benjamin/Cummings Publishing Company, Inc., 1994
8. Pacheco, S. P. Parallel programming with MPI. Morgan Kaufmann Publishers, San Francisco. 1997.

IV.2. ИНФОРМАЦИОННОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.

Информационные ресурсы сети Интернет

1. Информационно-аналитические материалы по параллельным вычислениям (http: //www. parallel. ru)

В преподавании курса «Алгоритмические языки и программирование» используются следующие формы:

- Образовательный процесс по дисциплине строится на основе комбинации образовательных технологий.
- Интегральную модель образовательного процесса по дисциплине формируют технологии методологического уровня:
  - модульно-рейтинговое обучение;
  - контекстное обучение;
  - технология поэтапного формирования умственных действий;
  - технология развивающего обучения;
  - элементы технологии развития критического мышления.

Реализация данной модели предполагает использование следующих технологий стратегического уровня, задающих организационные формы взаимодействия субъектов образовательного процесса, осуществляемых с использованием определенных тактических процедур:

- лекционные (вводная лекция, информационная лекция, обзорная лекция, лекция-консультация, лекция-визуализация, проблемная лекция);
- лабораторные работы (углубление знаний, полученных на теоретических занятиях, интерактивное выполнение, разработка, тестирование и отладка программ по темам лабораторных работ);
- тренинговые (формирование определенных умений и навыков, формирование алгоритмического мышления);
- активизации познавательной деятельности (приемы технологии развития критического мышления через чтение и письмо, работа с литературой, подготовка отчетов по лабораторным работам);
- самоуправления (самостоятельная работа студентов, самостоятельное изучение материала).

Рекомендуется использование информационных технологий при организации коммуникации со студентами для представления информации, выдачи рекомендаций и консультирования по оперативным вопросам (электронная почта), при проведения контроля выполнения лабораторных работ, использование мультимедиа-средств при проведении лекционных занятий.

IV.3. Материально-техническое обеспечение дисциплины

Для проведения дисциплины необходимо иметь:

Компьютерные классы, современные компьютеры которых объединены в локальную сеть.

Необходимое лицензионное программное обеспечение.

Разработанные лабораторные работы, включающие в себя обучающие тексты, набор пошаговых инструкций, учебных задач и заданий, демонстрационный материал и тестовые задания, размещенные на магнитных носителях с программным обеспечением.

Библиотечный фонд, имеющий в достаточном количестве печатные пособия с методическими указания по выполнению лабораторных работ и контрольных заданий.