


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Вятский государственный университет»
(ВятГУ)

Колледж ВятГУ

УТВЕРЖДАЮ
Директор колледжа
 Вахрушева Л.В.
31.08. 2018 г.

рег. №3-09.02.07.51_2018_0012

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования

для специальности

09.02.07 Информационные системы и программирование

Форма обучения
очная

2018 г.

СОДЕРЖАНИЕ

	стр.
1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	16
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ В ХОДЕ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	18
ПРИЛОЖЕНИЕ 1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ	19

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования

1.1. Область применения рабочей программы

Рабочая программа учебной дисциплины является частью основной профессиональной образовательной программы в соответствии с ФГОС СПО и примерной образовательной программой (при наличии) по специальности 09.02.07 Информационные системы и программирование.

1.2. Место учебной дисциплины в структуре основной профессиональной образовательной программы:

«Основы алгоритмизации и программирования» – учебная дисциплина общепрофессионального цикла, обязательной части образовательной программы.

1.3. Цели и задачи учебной дисциплины – требования к результатам освоения учебной дисциплины:

В результате освоения учебной дисциплины обучающийся должен уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования.
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

В результате освоения учебной дисциплины обучающийся должен знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

1.4. Формируемые компетенции

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов по очной форме обучения
Максимальная учебная нагрузка (всего)	138
Обязательная аудиторная учебная нагрузка (всего)	102
в том числе:	
теоретическое обучение	26
лабораторные занятия	70
практические работы	6
Промежуточная аттестация	9
Самостоятельная работа обучающегося (всего)	26
Консультации	1
Форма промежуточной аттестации – экзамен	

2.2. Тематический план учебной дисциплины «Основы алгоритмизации и программирования»

Название разделов / тем учебной дисциплины	Вид учебной работы	Объем часов			Уровень освоения
		Очная форма обучения	Заочная форма обучения	Заочная форма обучения с использованием ДОТ	
1	2	3	4	5	6
Тема 1. Языки программирования	Теоретическое обучение	4	-	-	1
	Практические работы	-	-	-	
	Самостоятельная работа обучающихся	2	-	-	
Тема 2. Типы данных	Теоретическое обучение	2	-	-	2
	Практические работы	-	-	-	
	Самостоятельная работа обучающихся	-	-	-	
Тема 3. Операторы языка программирования	Теоретическое обучение	2	-	-	2
	Лабораторные работы	20	-	-	
	Практические работы	4			
	Самостоятельная работа обучающихся	10	-	-	
Тема 4. Процедуры и	Теоретическое обучение	2	-	-	2

функции	Лабораторные работы	4	-	-	
	Самостоятельная работа обучающихся	-	-	-	
Тема 5. Структуризация в программировании	Теоретическое обучение	2	-	-	2
	Практические работы	-	-	-	
	Самостоятельная работа обучающихся	-	-	-	
Тема 6. Модульное программирование	Теоретическое обучение	2	-	-	2
	Лабораторные работы	4	-	-	
	Самостоятельная работа обучающихся	-	-	-	
Тема 7. Указатели	Теоретическое обучение	2	-	-	2
	Лабораторные работы	8	-	-	
	Самостоятельная работа обучающихся	4	-	-	
Тема 8. Основные принципы объектно-ориентированного программирования (ООП)	Теоретическое обучение	6	-	-	2
	Лабораторные работы	6	-	-	
	Самостоятельная работа обучающихся	-	-	-	
Тема 9. Интегрированная среда разработчика	Теоретическое обучение	4	-	-	2
	Практические работы	2	-	-	
	Лабораторные работы	6			
	Самостоятельная работа обучающихся	4	-	-	
Тема 10. Визуальное событийно-управляемое программирование	Теоретическое обучение	-	-	-	2
	Лабораторные работы	6	-	-	
	Самостоятельная работа обучающихся	-	-	-	
Тема 11. Разработка оконного приложения	Теоретическое обучение	-	-	-	2
	Лабораторные работы	10	-	-	
	Самостоятельная работа обучающихся	4	-	-	
Тема 12. Этапы разработки приложений	Теоретическое обучение	-	-	-	2
	Лабораторные работы	4	-	-	
	Самостоятельная работа обучающихся	-	-	-	
Тема 13. Иерархия классов	Теоретическое обучение	2	-	-	2
	Лабораторные работы	2	-	-	
	Самостоятельная работа обучающихся	-	-	-	
Экзамен		9	-	-	3
Консультация		1	-	-	
	Итого	138	-	-	

2.4. Содержание разделов / тем учебной дисциплины

Тема 1. Языки программирования

Содержание учебного материала:

Развитие языков программирования.

Обзор языков программирования. Области применения языков программирования.

Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.

Жизненный цикл программы. Программа. Программный продукт и его характеристики.

Основные этапы решения задач на компьютере.

Самостоятельная работа: проработка конспекта лекции, подготовка к опросу.

Формы текущего контроля по теме: устный опрос.

Вопросы для подготовки к текущей аттестации по теме:

1. Развитие языков программирования.
2. Обзор языков программирования.
3. Области применения языков программирования.
4. Стандарты языков программирования.
5. Среда проектирования.
6. Компиляторы и интерпретаторы.
7. Жизненный цикл программы.
8. Программа.
9. Программный продукт и его характеристики.
10. Основные этапы решения задач на компьютере.

Тема 2. Типы данных

Содержание учебного материала:

Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.

Формы текущего контроля по теме: письменный опрос.

Вопросы для подготовки к текущей аттестации по теме:

1. Этапы развития ЭВМ.
2. Классификация ЭВМ по принципу действия.
3. Поколения ЭВМ.
4. Классификация ЭВМ по назначению.
5. Классификация ЭВМ по функциональным возможностям.

Тема 3. Операторы языка программирования

Содержание учебного материала

Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.

Условный оператор. Оператор выбора.

Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.

Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.

Структурированный тип данных – множество. Операции над множествами.

Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа.

Практические работы (практические занятия):

1. Знакомство со средой программирования.
2. Алгоритмические структуры.

Лабораторные работы (лабораторные занятия):

1. Составление программ линейной структуры.
2. Составление программ разветвляющейся структуры.

3. Составление программ циклической структуры
4. Обработка одномерных массивов.
5. Обработка двумерных массивов.
6. Работа со строками.
7. Работа с данными типа множество.
8. Файлы последовательного доступа.
9. Типизированные файлы.
10. Нетипизированные файлы.

Самостоятельная работа: проработка конспекта лекции.

Формы текущего контроля по теме: практическая работа, лабораторная работа.

Вопросы для подготовки к текущей аттестации по теме:

1. Операции и выражения.
2. Правила формирования и вычисления выражений.
3. Структура программы.
4. Ввод и вывод данных.
5. Оператор присваивания.
6. Составной оператор.
7. Условный оператор.
8. Оператор выбора.
9. Цикл с постусловием.
10. Цикл с предусловием.
11. Цикл с параметром.
12. Вложенные циклы.
13. Массивы.
14. Двумерные массивы.
15. Строки.
16. Стандартные процедуры и функции для работы со строками.
17. Структурированный тип данных – множество.
18. Операции над множествами.
19. Комбинированный тип данных – запись.
20. Файлы последовательного доступа.
21. Файлы прямого доступа.

Тема 4. Процедуры и функции

Содержание учебного материала:

Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.

Рекурсия. Программирование рекурсивных алгоритмов.

Лабораторные работы (лабораторные занятия):

1. Организация процедур.
2. Организация функций.
3. Применение рекурсивных функций.

Формы текущего контроля по теме: тест, лабораторная работа.

Вопросы для подготовки к текущей аттестации по теме:

1. Общие сведения о подпрограммах.
2. Определение и вызов подпрограмм.
3. Область видимости и время жизни переменной.
4. Механизм передачи параметров.
5. Организация функций.
6. Рекурсия.
7. Программирование рекурсивных алгоритмов.

Тема 5. Структуризация в программировании

Содержание учебного материала:

Основы структурного программирования. Методы структурного программирования.

Формы текущего контроля по теме: устный опрос.

Вопросы для подготовки к текущей аттестации по теме:

1. Основы структурного программирования.
2. Методы структурного программирования.

Тема 6. Модульное программирование

Содержание учебного материала:

Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.

Стандартные модули.

Лабораторные работы (лабораторные занятия):

1. Программирование модуля.
2. Создание библиотеки подпрограмм.

Формы текущего контроля по теме: разработка ментальной карты, лабораторная работа.

Тема 7. Указатели

Содержание учебного материала:

Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.

Структуры данных на основе указателей.

Лабораторные работы (лабораторные занятия):

1. Задача о стеке.
2. Использование указателей для организации связанных списков.

Самостоятельная работа: проработка конспекта лекции.

Формы текущего контроля по теме: лабораторная работа.

Вопросы для подготовки к текущей аттестации по теме:

1. Указатели.
2. Описание указателей.
3. Основные понятия и применение динамически распределяемой памяти.
4. Создание и удаление динамических переменных.
5. Структуры данных на основе указателей.
6. Задача о стеке.

Тема 8. Основные принципы объектно-ориентированного программирования (ООП)

Содержание учебного материала:

История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.

Основные принципы ООП: инкапсуляция, наследование, полиморфизм.

Классы объектов. Компоненты и их свойства.

Событийно-управляемая модель программирования. Компонентно-ориентированный подход.

Лабораторные работы (лабораторные занятия):

1. Классы ООП: виды, назначение, свойства, методы, события.
2. Объявления класса.
3. Создание наследованного класса.

Формы текущего контроля по теме: устный опрос, лабораторная работа.

Вопросы для подготовки к текущей аттестации по теме:

1. История развития ООП.
2. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.

3. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
4. Классы объектов.
5. Компоненты и их свойства.
6. Событийно-управляемая модель программирования.
7. Компонентно-ориентированный подход.

Тема 9. Интегрированная среда разработчика

Содержание учебного материала:

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. Выполнение проекта. Настройка среды и параметров проекта.

Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.

Настройка среды и параметров проекта.

Лабораторные работы (лабораторные занятия):

1. Создание проекта с использованием компонентов для работы с текстом.
2. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.

Практические работы (практические занятия):

1. Изучение интегрированной среды разработчика. Настройка среды и параметров проекта.

Самостоятельная работа: проработка конспекта лекции.

Формы текущего контроля по теме: лабораторная работа, практическая работа.

Вопросы для подготовки к текущей аттестации по теме:

1. Требования к аппаратным и программным средствам интегрированной среды разработчика.
2. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты.
3. Форма и размещение на ней управляющих элементов.
4. Панель компонентов и их свойства.
5. Окно кода проекта.
6. Состав и характеристика проекта.
7. Выполнение проекта.
8. Настройка среды и параметров проекта.
9. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.
10. Настройка среды и параметров проекта.

Тема 10. Визуальное событийно-управляемое программирование

Содержание учебного материала:

Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.

Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.

События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.

Лабораторные работы (лабораторные занятия):

1. Создание процедур на основе событий.

2. Создание проекта с использованием кнопочных компонентов.
3. Создание проекта с использованием компонентов стандартных диалогов и системы меню.

Формы текущего контроля по теме: лабораторная работа.

Вопросы для подготовки к текущей аттестации по теме:

1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
2. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.
3. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.

Тема 11. Разработка оконного приложения

Содержание учебного материала:

Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Разработка игрового приложения.

Лабораторные работы (лабораторные занятия):

1. Создание процедур на основе событий.
2. Создание проекта с использованием кнопочных компонентов.
3. Создание проекта с использованием компонентов стандартных диалогов и системы меню.

Формы текущего контроля по теме: лабораторная работа.

Самостоятельная работа: проработка конспекта лекции.

Вопросы для подготовки к текущей аттестации по теме:

1. Разработка интерфейса приложения.
2. Создание проекта с использованием кнопочных компонентов.
3. Создание проекта с использованием компонентов стандартных диалогов и системы меню.

Тема 12. Этапы разработки приложений

Содержание учебного материала:

Разработка приложения. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Тестирование, отладка приложения.

Лабораторные работы (лабораторные занятия):

1. Разработка функциональной схемы работы приложения.
2. Разработка оконного приложения с несколькими формами.
3. Разработка игрового приложения.
4. Тестирование, отладка приложения.

Самостоятельная работа: проработка конспекта лекции.

Вопросы для подготовки к текущей аттестации по теме:

1. Разработка приложения.
2. Проектирование объектно-ориентированного приложения.
3. Создание интерфейса пользователя.
4. Тестирование, отладка приложения.

Тема 13. Иерархия классов

Содержание учебного материала:

Классы ООП: виды, назначение, свойства, методы, события. Перегрузка методов. Тестирование и отладка приложения. Решение задач.

Лабораторные работы (лабораторные занятия):

1. Программирование приложений.

2. Перегрузка методов.

Формы текущего контроля по теме: лабораторная работа.

Самостоятельная работа: проработка конспекта лекции.

Вопросы для подготовки к текущей аттестации по теме:

1. Классы ООП: виды, назначение, свойства, методы, события.
2. Перегрузка методов.
3. Тестирование и отладка приложения.
4. Решение задач.

Методические указания для обучающихся по освоению дисциплины

Успешное освоение учебной дисциплины предполагает активное, творческое участие обучающегося на всех этапах ее освоения путем планомерной, повседневной работы. Обучающийся обязан посещать лекции и практические, лабораторные занятия, получать консультации преподавателя и выполнять самостоятельную работу.

Выбор методов и средств обучения, образовательных технологий осуществляется преподавателем исходя из необходимости достижения обучающимися планируемых результатов освоения дисциплины, а также с учетом индивидуальных возможностей обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья.

Организация учебного процесса предусматривает применение инновационных форм учебных занятий, развивающих у обучающихся навыки командной работы, межличностной коммуникации, принятия решений, лидерские качества (включая, при необходимости, проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей).

Изучение дисциплины следует начинать с проработки настоящей рабочей программы, методических указаний и разработок, указанных в программе, особое внимание уделить целям, задачам, структуре и содержанию дисциплины.

Главной задачей каждой лекции является раскрытие сущности темы и анализ ее основных положений. Содержание лекций определяется настоящей рабочей программой дисциплины.

Лекции – это систематическое устное изложение учебного материала. На них обучающийся получает основной объем информации по каждой конкретной теме. Лекции обычно носят проблемный характер и нацелены на освещение наиболее трудных и дискуссионных вопросов, кроме того они способствуют формированию у обучающихся навыков самостоятельной работы с научной литературой.

Предполагается, что обучающиеся приходят на лекции, предварительно проработав соответствующий учебный материал по источникам, рекомендуемым программой. Часто обучающимся трудно разобраться с дискуссионными вопросами, дать однозначный ответ. Преподаватель, сравнивая различные точки зрения, излагает свой взгляд и нацеливает их на дальнейшие исследования и поиск научных решений. После лекции желательно вечером перечитать и закрепить полученную информацию, тогда эффективность ее усвоения значительно возрастает. При работе с конспектом лекции необходимо отметить материал, который вызывает затруднения для понимания, попытаться найти ответы на затруднительные вопросы, используя предлагаемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь за помощью к преподавателю.

Целью практических и лабораторных занятий является проверка уровня понимания обучающимися вопросов, рассмотренных на лекциях и в учебной литературе, степени и качества усвоения материала; применение теоретических знаний в реальной практике решения задач; восполнение пробелов в пройденной теоретической части курса и оказания помощи в его освоении.

Практические, лабораторные занятия в равной мере направлены на совершенствование индивидуальных навыков решения теоретических и прикладных задач, выработку навыков интеллектуальной работы, а также ведения дискуссий. Конкретные пропорции разных видов работы в группе, а также способы их оценки определяются преподавателем, ведущим занятия.

На практических, лабораторных занятиях под руководством преподавателя обучающиеся обсуждают дискуссионные вопросы, отвечают на вопросы тестов, закрепляя приобретенные знания, выполняют практические, лабораторные задания и т.п. Для успешного проведения практического, лабораторного занятия обучающемуся следует тщательно подготовиться.

Основной формой подготовки обучающихся к практическим, лабораторным занятиям является самостоятельная работа с учебно-методическими материалами, научной литературой, статистическими данными и т.п.

Изучив конкретную тему, обучающийся может определить, насколько хорошо он в ней разобрался. Если какие-то моменты остались непонятными, целесообразно составить список вопросов и на занятии задать их преподавателю. Практические, лабораторные занятия предоставляют студенту возможность творчески раскрыться, проявить инициативу и развить навыки публичного ведения дискуссий и общения, сформировать определенные навыки и умения и т.п.

Самостоятельная работа студентов включает в себя выполнение различного рода заданий (изучение учебной и научной литературы, материалов лекций, систематизацию прочитанного материала, подготовку контрольной работы, решение задач и т.п.), которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По каждой теме учебной дисциплины преподаватель предлагает обучающимся перечень заданий для самостоятельной работы. Самостоятельная работа по учебной дисциплине может осуществляться в различных формах (например, подготовка докладов; написание рефератов; другие).

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны исполняться самостоятельно либо группой и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению.

Каждую неделю рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Результатом самостоятельной работы должно стать формирование у обучающегося определенных знаний, умений, компетенций.

Система оценки качества освоения учебной дисциплины включает текущий контроль успеваемости, промежуточную аттестацию.

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины, промежуточная аттестация обучающихся - оценивание промежуточных и окончательных результатов обучения по дисциплине.

Процедура оценивания результатов освоения учебной дисциплины осуществляется на основе действующего Положения об организации текущего контроля успеваемости и промежуточной аттестации обучающихся ВятГУ.

Для приобретения требуемых знаний, умений и высокой оценки по дисциплине обучающимся необходимо выполнять все виды работ своевременно в течение семестра.

3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация учебной дисциплины требует наличия лаборатории программирования баз данных и учебной аудитории для лекционных занятий.

Основное оборудование лаборатории программирования баз данных:

- рабочие места обучающихся;
- автоматизированные рабочие места обучающихся;
- рабочее место преподавателя;
- передвижная маркерная доска.
- мультимедийный проектор;
- экран;
- ноутбук;
- сервер.

Учебно-наглядные пособия:

- комплект плакатов «Основы информатики».

Программное обеспечение:

- Windows Professional;
- Office Professional Plus;
- Microsoft Visual Studio Community;
- SQLServer Express Edition;
- SQLServer Management Studio;
- MySQLInstaller for Windows;
- AMPPS;
- Notepad++;
- Atom;
- Git;
- Microsoft Visio Professional;
- Microsoft Project.

Оборудование учебной аудитории для лекционных занятий:

- рабочее место преподавателя;
- рабочие места обучающихся;
- учебная доска;
- экран;
- мультимедийный проектор;
- ноутбук;

Программное обеспечение:

- Windows Professional;
- Office Professional Plus.

3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, дополнительной литературы, Интернет-ресурсов

Основная литература:

1. Семакин, Игорь Геннадьевич. Основы алгоритмизации и программирования [Электронный ресурс] / И. Г. Семакин, А. П. Шестаков. - Москва: Академия, 2017. - 304 с.

2. Семакин, Игорь Геннадьевич. Основы алгоритмизации и программирования [Текст]: учебник / И. Г. Семакин, А. П. Шестаков. - Москва: Академия, 2018. - 300 с.

Дополнительная литература:

1. Лубашева, Т. В. Основы алгоритмизации и программирования [Электронный ресурс]: учебное пособие / Т.В. Лубашева. - Минск: РИПО, 2016. - 378 с.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Раздел официального сайта ВятГУ, содержащий описание образовательной программы [Электронный ресурс] /-Режим доступа: <https://www.vyatsu.ru/nash-universitet/obrazovatel'naya-deyatel-nost/kolledzh/09-02-07-informatsionnyie-sistemyi-i-programmirova.html>

Перечень электронно-библиотечных систем (ресурсов) и баз данных для самостоятельной работы

Используемые сторонние электронные библиотечные системы (ЭБС):

1. ЭБС «Научная электронная библиотека eLIBRARY» (<http://elibrary.ru/defaultx.asp>)
2. ЭБС «Издательства Лань» (<http://e.lanbook.com/>)
3. ЭБС «Университетская библиотека online» (www.biblioclub.ru)
4. ЭБС «ЮРАЙТ» (<http://biblio-online.ru>)
5. ЭБС «Академия» (<http://www.academia-moscow.ru/>)
6. Свободный каталог периодики библиотек России (<http://ucpr.arbicon.ru/>)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

- Windows Professional;
- Office Professional Plus;
- Microsoft Visual Studio Community;
- SQLServer Express Edition;
- SQLServer Management Studio;
- MySQLInstaller for Windows;
- AMPPS;
- Notepad++;
- Atom;
- Git;
- Microsoft Visio Professional;
- Microsoft Project.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ В ХОДЕ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Результаты обучения	Формы и методы контроля для оценки результатов обучения
<p>Освоенные умения:</p> <ul style="list-style-type: none">– разрабатывать алгоритмы для конкретных задач;– использовать программы для графического отображения алгоритмов;– определять сложность работы алгоритмов;– работать в среде программирования.– реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;– оформлять код программы в соответствии со стандартом кодирования;– выполнять проверку, отладку кода программы. <p>Усвоенные знания:</p> <ul style="list-style-type: none">– понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;– эволюцию языков программирования, их классификацию, понятие системы программирования;– основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;– подпрограммы, составление библиотек подпрограмм;– объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.	<p>Экзамен в форме</p> <ul style="list-style-type: none">- устного опроса,- выполнения практических заданий

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ
АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
«Основы алгоритмизации и программирования»**

1. Общие положения

Формы и процедуры промежуточной аттестации по дисциплине разрабатываются преподавателями и доводятся до сведения обучающихся в течение первых двух месяцев от начала обучения.

Промежуточный контроль по учебной дисциплине осуществляется в форме экзамена.

Виды заданий промежуточной аттестации: устный ответ, практическое задание.

2. Методические материалы, определяющие процедуры оценивания результатов обучения

2.1. Промежуточная аттестация в форме экзамена

Цель процедуры:

Целью промежуточной аттестации по учебной дисциплине является оценка уровня усвоения обучающимися знаний и освоения умений в результате изучения учебной дисциплины.

Субъекты, на которые направлена процедура:

Процедура оценивания должна охватывать всех без исключения обучающихся, осваивающих дисциплину. В случае, если обучающийся не прошел процедуру без уважительных причин, то он считается имеющим академическую задолженность.

Период проведения процедуры:

Процедура оценивания проводится по окончании изучения дисциплины в период промежуточной аттестации, в соответствии с календарным учебным графиком.

Требования к помещениям материально-техническим средствам для проведения процедуры:

Требования к лаборатории для проведения процедуры и необходимости специализированных материально-технических средств определяются преподавателем, ведущим дисциплину.

Требования к кадровому обеспечению проведения процедуры:

Процедуру проводит преподаватель, ведущий дисциплину.

Требования к фонду оценочных средств:

До начала проведения процедуры преподавателем разрабатывается фонд оценочных средств для оценки знаний и умений, который включает примерные вопросы и задания, из перечня которых формируются экзаменационные билеты. Экзаменационные билеты рассматриваются на соответствующих цикловых комиссиях и утверждаются заместителем директора колледжа по учебной работе. Количество вопросов в билете определяется преподавателем самостоятельно в зависимости от вида заданий, но не менее двух. Количество экзаменационных билетов, как правило, превышает количество обучающихся, проходящих процедуру промежуточной аттестации в форме экзамена.

Описание проведения процедуры:

Каждому обучающемуся при предъявлении зачетной книжки выдается экзаменационный билет. После получения экзаменационного билета и подготовки ответов, обучающийся должен в меру имеющихся знаний и умений выполнить предложенные задания в установленное преподавателем время. Продолжительность проведения экзамена определяется из расчета 0,3 часа на каждого обучающегося.

Шкалы оценки результатов проведения процедуры:

Результаты проведения экзамена оцениваются преподавателем с применением четырехбалльной шкалы в соответствии с критериями оценки.

3. Контроль и оценка образовательных результатов

Для контроля и оценки образовательных результатов по учебной дисциплине разрабатываются фонды оценочных средств, которые позволяют оценить все предусмотренные рабочей программой умения и знания.

3.1. Показатели оценки образовательных результатов

Образовательные результаты (знания, умения)	Показатели оценки результата
– разрабатывать алгоритмы для конкретных задач;	Решение задач с помощью конкретных алгоритмов.
– использовать программы для графического отображения алгоритмов;	Отображение графического представления алгоритмов.
– определять сложность работы алгоритмов;	Определение сложности работы алгоритмов
– работать в среде программирования.	Выбор среды программирования.
– реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;	Разработка построенных алгоритмов в виде программ на конкретном языке программирования.
– оформлять код программы в соответствии со стандартом кодирования;	Оформление кода программы в соответствии со стандартом кодирования
– выполнять проверку, отладку кода программы.	Отладка кода программы.
– понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;	Формулирование понятия алгоритмизации, свойств алгоритмов, общих принципов построения алгоритмов, основных алгоритмических конструкций.
– эволюцию языков программирования, их классификацию, понятие системы программирования;	Описание эволюции языков программирования, их классификацию, понятие системы программирования.
– основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;	Описание основных элементов языка, структуры программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти
– подпрограммы, составление библиотек подпрограмм;	Выбор и разработка подпрограмм, составление библиотек подпрограмм
– объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.	Описание объектно-ориентированной модели программирования, основных принципов объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

3.2. Перечень вопросов для контроля знаниевых образовательных результатов

Проверяемые образовательные результаты (знания)	Примерные вопросы для контроля в соответствии с уровнем освоения
– понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;	1. Развитие языков программирования. 2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.
– эволюцию языков программирования, их классификацию, понятие системы программирования;	3. Жизненный цикл программы. Программа. Программный продукт и его характеристики. 4. Основные этапы решения задач на компьютере. 5. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.
– основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;	6. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор. 7. Условный оператор. Оператор выбора.
– подпрограммы, составление библиотек подпрограмм;	8. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.
– объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.	9. Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками. 10. Структурированный тип данных – множество. Операции над множествами. 11. Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа 12. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций. 13. Рекурсия. Программирование рекурсивных алгоритмов. 14. Основы структурного программирования. Методы структурного программирования. 15. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы. 16. Стандартные модули. 17. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных. 18. Структуры данных на основе указателей. 19. Задача о стеке. 20. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. 21. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. 22. Классы объектов. Компоненты и их свойства. 23. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. 24. Требования к аппаратным и программным средствам интегрированной среды разработчика.

	<p>25.Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.</p> <p>26.Панель компонентов и их свойства. Окно кода проекта.</p> <p>27.Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.</p> <p>28.Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.</p> <p>29.Настройка среды и параметров проекта.</p> <p>30.Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.</p> <p>31.Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.</p> <p>32.События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.</p> <p>33.Разработка функционального интерфейса приложения. Создание интерфейса приложения.</p> <p>34.Разработка функциональной схемы работы приложения.</p> <p>35.Разработка приложения.</p> <p>36.Проектирование объектно-ориентированного приложения.</p> <p>37.Создание интерфейса пользователя.</p> <p>38.Тестирование, отладка приложения.</p> <p>39.Классы ООП: виды, назначение, свойства, методы, события.</p> <p>40.Перегрузка методов.</p>
--	---

3.2.1. Перечень заданий для контроля умениевых образовательных результатов

Проверяемые образовательные результаты (умения)	Примерные практические задания для контроля в соответствии с уровнем освоения
– разрабатывать алгоритмы для конкретных задач;	1. Дано натуральное число N. Найти сумму его четных делителей.
– использовать программы для графического отображения алгоритмов;	2. Дано натуральное число. Определить, какая цифра встречается чаще: 0 или 9.
– определять сложность работы алгоритмов;	3. Найти сумму $-12 + 22 - 32 + 42 - 52 + \dots + 402$. (Условный оператор не использовать)
– работать в среде программирования.	4. Найти все трехзначные числа, которые делятся на n или содержат цифру n (n вводится с клавиатуры).
– реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;	5. Дано натуральное число n . Вычислить: $P = 2 \times 4 \times 6 \times \dots \times (2n)$
– оформлять код программы в соответствии со стандартом кодирования;	6. Дано вещественное число a. Напечатать все значения n, при которых: $1+1/2+1/3+\dots+1/n < a$
	7. Дано натуральное число. Определить, есть ли в нем цифра 3.
	8. Вычислить сумму: $2/3+3/4+4/5+\dots+10/11$.
	9. Среди чисел 1, 4, 9, 16, 25, . . . найти первое число,

<p>– выполнять проверку, отладку кода программы.</p>	<p>большее заданного числа N.</p> <p>10. Вычислить: $(1+\sin 0,1) \cdot (1+\sin 0,2) \cdot \dots \cdot (1+\sin 9,9) \cdot (1+\sin 10)$.</p> <p>11. Вычислить: $y=1!+2!+3!+\dots+n!$ ($n>1$).</p> <p>12. Дано натуральное число. Определить, является ли разность его максимальной и минимальной цифр четным числом.</p> <p>13. Даны два натуральных числа. Выяснить, в каком из них сумма цифр больше (использовать процедуру).</p> <p>14. Даны шесть различных натуральных чисел. Определить максимальное из них (использовать функцию нахождения максимума из двух).</p> <p>15. Даны стороны двух треугольников. Найти сумму их площадей (использовать процедуру нахождения площади треугольника).</p> <p>16. Вычислить значение выражения $(2*5!+3*8!)/(6!+4!)$, используя функцию для вычисления $n!$</p> <p>17. Даны 3 натуральных числа. Определить, у какого из них среднее арифметическое цифр больше (использовать функцию или процедуру).</p> <p>18. Дан одномерный массив целых чисел. Найти максимальный элемент массива и поменять его местами с первым элементом.</p> <p>19. Известно, что в одномерном массиве имеются элементы, равные 5. Определить номер первого из них. (Условный оператор не использовать).</p> <p>20. Дан одномерный массив из четного числа элементов. Поменять местами его первый элемент со вторым, третий с четвертым и т.д.</p> <p>21. Дан массив, содержащий 10 трехзначных чисел. Заменить каждый элемент массива разностью максимальной и минимальной из его цифр.</p> <p>22. Дан одномерный массив целых чисел. Напечатать все элементы, следующие за последним, оканчивающиеся цифрой "7". Если элементов, оканчивающихся цифрой "7", в массиве нет, то ни один элемент не должен быть напечатан.</p> <p>23. Дан двумерный массив размером n на n, заполненный целыми числами. Все его отрицательные элементы записать в первый одномерный массив, все его четные элементы - во второй, а нечетные - в третий.</p> <p>24. Дан двумерный массив целых чисел. В каждой его строке найти сумму отрицательных элементов.</p> <p>25. Дан двумерный массив целых чисел. Определить минимальный номер столбца, состоящего только из элементов, больших числа 10 (считать, что такой столбец обязательно есть).</p> <p>26. Дан двумерный массив. Найти строку с максимальной суммой элементов. Дополнительный массив не использовать.</p>
--	--

3.2.2. Критерии оценки образовательных результатов

1. Шкала оценки устных ответов

Критерии	Качественная оценка образовательных результатов.	
	балл (отметка)	вербальный аналог
Тема раскрыта в полном объеме, высказывания связные и логичные, использована научная лексика, приведены примеры, сделаны выводы. Ответы на вопросы даны в полном объеме или вопросы отсутствуют.	5	отлично
Тема раскрыта не в полном объеме, высказывания в основном связные и логичные, использована научная лексика, приведены примеры, сделаны выводы. Ответы на вопросы сигнализируют о наличии проблемы в понимании темы.	4	хорошо
Тема раскрыта недостаточно, высказывания несвязные и нелогичные. Научная лексика не использована, примеры не приведены, выводы отсутствуют. Ответы на вопросы в значительной степени зависят от помощи со стороны преподавателя.	3	удовлетворительно
Тема не раскрыта. Логика изложения, примеры, выводы и ответы на вопросы отсутствуют.	2	не удовлетворительно

2. Шкала оценки модельных ответов

Критерии	Качественная оценка образовательных результатов	
	балл (отметка)	вербальный аналог
Задание выполнено в соответствии с модельным ответом	5	отлично
В задании допущен один-два недочета и (или) одна ошибка	4	хорошо
В задании допущено несколько недочётов и две ошибки	3	удовлетворительно
В задании допущено несколько недочетов и более двух ошибок	2	не удовлетворительно