

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Худин Александр Николаевич

Должность: Ректор

Дата подписания: 31.08.2021 12:07:44

Уникальный программный ключ:

08303ad8de1c60b987361de7085acb509ac3da143f415362ffaf0ee37e73fa19

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курский государственный университет»

Колледж коммерции, технологий и сервиса

УТВЕРЖДЕНО
протокол заседания
ученого совета от 01.11.2021 г., № 3

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования



Курск 2021

Рабочая программа учебной дисциплины разработана на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования (далее – СПО) **09.02.07 Информационные системы и программирование**

Организация – разработчик: ФГБОУ ВО «Курский государственный университет».

Разработчик:

Бобрышева В.В. – преподаватель колледжа коммерции, технологий и сервиса ФГБОУ ВПО «Курский государственный университет».

СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	стр. 3
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	5
3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	16
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	18

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования

1.1. Область применения программы

Рабочая программа учебной дисциплины является частью ППССЗ в соответствии с ФГОС по специальности СПО **09.02.07 Информационные системы и программирование**.

Рабочая программа учебной дисциплины может быть использована в дополнительном профессиональном образовании (в программах повышения квалификации и переподготовки).

1.2. Место дисциплины в структуре программы подготовки специалистов среднего звена:

дисциплина входит профессиональный цикл

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

Процесс изучения учебной дисциплины направлен на формирование следующих компетенций:

ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 2	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
ОК 4	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
ОК 9	Использовать информационные технологии в профессиональной деятельности.
ОК 10	Пользоваться профессиональной документацией на государственном языке.
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием.
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств.
ПК 1.4	Выполнять тестирование программных модулей.
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода.
ПК 2.4	Осуществлять разработку текстовых наборов и текстовых сценариев для программного обеспечения.
ПК 2.5	Производить инспектирование компонент программного обеспечения.

ния на предмет соответствия стандартам кодирования.

В результате освоения 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 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. Рекомендуемое количество часов на освоение программы дисциплины:

максимальной учебной нагрузки обучающегося 322 часов, в том числе:

обязательной аудиторной учебной нагрузки обучающегося 292 часов;

самостоятельной работы обучающегося 12 часов.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	322
Обязательная аудиторная учебная нагрузка (всего)	292
в том числе:	
лабораторные занятия	-
практические занятия	146
контрольные работы	-
Самостоятельная работа обучающегося (всего)	12
Оформление плана-конспекта	
Подготовка реферата	
Индивидуальные проектные задания	
Оформление отчета по практическим работам	
Консультации	6
Итоговая аттестация в форме <i>экзамена</i>	12

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

Наименование разделов и тем	Содержание учебного материала, лабораторные и практические работы, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов	Уровень освоения
1	2	3	4
Раздел 1. Основные понятия алгоритмизации		28	
Тема 1.1. Основные понятия алгоритмизации	Содержание	28	
	1 Понятие алгоритма, его свойства и виды. Критерии «хорошего» алгоритма. Способы описания алгоритмов: псевдокоды, блок – схема, программа.	10	1
	2 Базовые алгоритмические конструкции. Линейная, разветвляющаяся, циклическая. Блок- схема: основные элементы, правила составления.		2
	3 Различные комбинации алгоритмических конструкций. Текстовые данные. Математическая модель алгоритма.		2
	4 Классические алгоритмы Евклида. Алгоритм Евклида для целых чисел. Геометрический алгоритм Евклида.		2
	5 Алгоритмы решения нелинейных и линейных уравнений. Метод простых итераций для решения нелинейных систем. Метод Зейделя для решения нелинейных систем уравнений. Метод Ньютона для решения нелинейных систем уравнений. Метод секущих для решения нелинейных систем уравнений		2
	Практические занятия	16	
	1 №1. Составление блок-схем линейных алгоритмов	2	3
	2 №2. Составление блок-схем алгоритмов ветвления	2	
	3 №3. Составление блок-схем циклических алгоритмов	2	
	4 №4. Составление блок-схем алгоритмов усложненной структуры	2	
	5 №5. Составление блок-схем алгоритмов поиска минимального элемента.	2	
	6 №6. Составление блок-схем алгоритмов бинарного поиска	2	
	7 №7. Составление блок-схем алгоритмов сортировки массива простым выбором.	2	
	8 №8. Составление блок-схем алгоритмов сортировки массива методом обмена	2	

	Самостоятельная работа обучающихся Контрольная работа по теме «Основные понятия алгоритмизации»		2 ¹	
Раздел 2. Основы программирования			146	
Тема 2.1 Основные элементы процедурного языка	Содержание		33	
	1	История языков программирования. Генеалогическое древо C#. Преимущества и недостатки языка C#.	20	1
	2	Платформа .NET Framework. Назначение, составные элементы, тонкости компиляции. Обзор интегрированной среды разработки Visual Studio.Net.		2
	3	Создание, сохранение и закрытие проектов и решений. Структура программы. Точка входа.		
	4	Базовый синтаксис C#. Пространство имен. Классы.		
	5	Типы данных. Значимые и ссылочные типы данных. Преобразование типов: явное и неявное.		
	6	Переменные. Объявления переменных и их инициализация. Определение, правила именования. Область действия и время существования переменных.		
	7	Константы. Суффиксы целых и вещественных констант. Определение, виды и правила записи в программе. Escape - последовательности.		
	8	Понятие выражения. Математические операторы. Старшинство операторов.		
	9	Математические функции (класс Math) Поля. Методы. Примеры.		
	10	Ввод-вывод данных. Оператор присваивания. Общий вид оператора присваивания. Способы ввода и вывода информации.		
	Практические занятия		10	
	1	№9. Знакомство со средой Visual Studio.Net.	10	3
2	№10. Создание консольного приложения.			

¹ в т.ч. 1 час консультаций

	3	№11. Организация простейшего ввода - вывода		
	4	№12. Разработка линейных программ. Операции и выражения.		
	5	№13. Использование математических функций класса Math		
	Самостоятельная работа обучающихся Решение задач по образцу по теме: «Операции и выражения»		3 ²	
Тема 2.2. Управляющие структуры	Содержание		20	
	1	Операторы отношения. Оператор if-else. Назначение правила записи. Проверка простых и сложных условий. Вложенные условные операторы.	12	2
	2	Использование поразрядных и условных логических операторов. Отличие в использовании операторов & и &&, и .		
	3	Тернарный оператор. Правила использования.		
	4	Механизм получения случайного числа. Источники случайных чисел. Качественные требования, предъявляемые к случайным числам.		
	5	Оператор выбора switch – case. Назначение и правила записи.		
	6	Операторы перехода: break, goto, return. Назначение и правила записи.		
	Практические занятия		6	
	1	№14. Создание проектов разветвляющейся структуры	2	3
	2	№15. Создание проектов с использованием логических операторов	2	
	3	№16. Создание проектов с использованием оператора выбора switch - case	2	
	Самостоятельная работа обучающихся Решение задач по образцу по теме «Управляющие структуры»		2	
Тема 2.3. Операторы цикла	Содержание		22	
	1	Цикл с предусловием Определение. Общий вид цикла	12	2

² в т.ч. 2 час консультаций

	2	Цикл с постусловием Определение. Общий вид цикла		
	3	Цикл с параметром Определение. Общий вид цикла		
	4	Получение таблицы значений Типизация. Строка таблицы значений. Изменение содержимого.		
	5	Нахождение суммы и произведения элементов ряда Рассмотрение примеров		
	6	Принудительный выход из цикла. Операторы break и continue. Бесконечные циклы.		
		Практические занятия	10	
	1	№17. Создание проектов с использованием циклического оператора while	10	3
	2	№18. Создание проектов с использованием циклического оператора do		
	3	№19. Создание проектов с использованием циклического оператора for		
	4	№20. Создание проектов с использованием break, goto, return		
	5	№21. Решение задач с применением циклов		
Тема 2.4. Структуры данных		Содержание	42	
	1	Массивы. Определение, виды. Объявление одномерного массива. Варианты инициализации	22	2
	2	Ввод и вывод одномерных массивов Способы ввода и вывода данных массива		
	3	Стандартные операции для работы с массивами. Заполнение случайными значениями и значениями по формуле		
	4	Нахождение суммы и произведения Рассмотрение примеров		
	5	Нахождение максимума (минимума) Рассмотрение примеров		
	6	Подсчет количества элементов, удовлетворяющих определенному условию Рассмотрение примеров		
	7	Обработка одномерных массивов: сортировка Рассмотрение примеров		

	8	Обработка одномерных массивов: поиск элементов Рассмотрение примеров		
	9	Цикл foreach Рассмотрение примеров		
	10	Двумерные массивы: объявление, ввод и вывод Рассмотрение примеров.		
	11	Работа с двумерными массивами по строкам и по столбцам. Рассмотрение примеров		
	Практические занятия		18	3
	1	№22. Создание проектов с использованием одномерных массивов		
	2	№23. Создание проектов с использованием двумерных массивов		
	3	№24. Создание проектов с использованием методов сортировки		
	4	№25. Создание проектов с использованием методов поиска		
	5	Создание проектов с использованием цикла foreach		
	6	№26. Создание проектов с использованием методов класса Array: Sort, Indexof, BinarySearch		
	7	№27. Создание проектов с использованием ступенчатых массивов		
	8	№28. Создание проектов с использованием массивов символов		
	9	№29. Создание проектов с использованием строк		
	Самостоятельная работа обучающихся Контрольная работа по теме «Структуры данных»		2	
Тема 2.5. Потоки. Файл	Содержание		29	
	1	Потоки. Понятие потока. Механизм буферизации. Классы библиотеки .Net для работы с потоками.	12	2
	2	Виды доступа к файлам. Объект FileStream. Примеры. Обнаружение изменений в положении потока. Конструкторы.		
	3	Классы StreamWriter и StreamReader Примеры. Конструкторы. Поля. Свойства. Методы		
	4	Основные операции при работе с файлами. Способы работы с текстовыми файлами. Примеры		

	5	Обработка текстовых файлов. Чтение и запись файла		
	6	Работа с каталогами и файлами. Команды для работы с файлами и каталогами		
	Практические занятия		16	3
	1	№30. Создание проектов с использованием class Random		
	2	№31. Создание проектов с использованием обработки исключительных ситуаций		
	3	№32. Создание проектов с использованием операторов checked и unchecked		
	4	№33. Создание проектов с использованием многопоточных приложений		
	5	№34. Создание проектов с использованием текстовых файлов		
	6	№35. Создание проектов с использованием двоичных файлов		
	7	№36. Создание проектов с использованием асинхронного ввода-вывода		
	8	№37. Создание проектов с использованием копирования файлов		
	Самостоятельная работа обучающихся Решение задач по образцу по теме «Потоки. Файлы»		1	
Раздел 3. Объектно-ориентированная модель программирования			136	
Тема 3.1 Основные правила объектно-ориентированного программирования	Содержание		60	2
	1	Понятие класса и объекта. Характеристики объекта: поля, свойства, методы, события	22	
	2	Основные принципы объектно-ориентированного программирования. Наследование, полиморфизм, инкапсуляция		
	3	Общая форма определения класса. Модификаторы доступа к элементам класса: public, private, protected, internal		
	4	Примеры создания классов. Классы. Объекты. Конструктор. Инициализаторы объектов		
	5	Переменные ссылочного типа и присваивания. Побочные эффекты множественных ссылок.		

	6	Метод. Вызов метода. Понятие, правила записи. Передача параметров по значению. Правило триединого соответствия параметров и аргументов: по количеству, типам и по порядку следования.		
	7	Создание методов, возвращающих значение. Способы размещения методов. Конструкторы.		
	8	Инкапсуляция как управление доступом к данным. Свойства класса: понятие, виды, правила записи.		
	9	Наследование и полиморфизм. Иерархия классов: понятие, преимущества. Синтаксис наследования.		
	10	Скрытие и перекрытие методов. Ключевые слова virtual, override. Вызов методов базового класса («родителя»): ключевое слово base.		
	11	Тонкости использования конструкторов в иерархически связанных между собой классах. Виды конструкторов. Примеры		
	Практические занятия		36	3
	1	№38. Решение задач на создание классов со свойствами.		
	2	№39. Создание проектов с использованием текстовых элементов управления и окон сообщений		
	3	№40. Создание проектов с использованием диалоговых окон		
	4	№41. Создание проектов с использованием кнопок		
	5	№42. Создание проектов с использованием компонентов для создания меню		
	6	№43. Создание проектов с использованием графических объектов		
	7	№44. Создание проектов с использованием многооконного интерфейса		
	8	№45. Создание проектов с использованием списков		
	9	№46. Создание проектов с использованием переключателей		
	10	№47. Создание проектов с использованием текстовых редакторов		
	11	№48. Создание проектов с использованием перегрузки методов		
	12	№49. Создание проектов с использованием рекурсивных методов		
	13	№50. Создание проектов с использованием метода Main		

	14	№51. Создание проектов с использованием методов с переменным количеством аргументов		
	15	№52. Создание классов, иерархически связанных между собой		
	16	№53. Создание проектов с использованием абстрактных классов		
	17	№54. Создание проектов с использованием виртуальных методов		
	18	№55. Создание проектов и использованием наследования		
	Самостоятельная работа обучающихся Подготовка реферата по теме «Способы реализации интерфейсов»		2	
Тема 3.2 Интерфейсы	Содержание		26	
	1	Интерфейсы. Назначение, правила описания. Способы реализации интерфейсов	14	2
	2	Работа с объектами через интерфейсы. Модификаторы доступа интерфейсов. Применение интерфейсов. Интерфейс по умолчанию		
	3	Операторы is и as. Восходящие и нисходящие приведения. Наследование в интерфейсах		
	4	Сходства и различия интерфейсов. Интерфейсы абстрактных и обычных классов		
	5	Стандартные интерфейсы .NET:IComparable Параметры типа. Примеры. Методы		
	6	Стандартные интерфейсы .NET:IClonable Параметры типа. Примеры. Методы		
	7	Стандартные интерфейсы .NET:IEnumerable Параметры типа. Примеры. Методы		
	Практические занятия		10	
	1	№56. Создание проектов с использованием интерфейса IComparable		3
	2	№57. Создание проектов с использованием интерфейса ICloneable		
	3	№58. Создание проектов с использованием интерфейса IEnumerable		
	4	№59. Создание проектов с использованием операторов is и as		
5	№60. Создание проектов с использованием наследования в интерфейсах			

	Самостоятельная работа обучающихся Подготовка реферата по теме «Способы реализации интерфейсов»	2 ³	
Тема 3.3 Структуры и перечисления	Содержание	18	
	1 Назначение и синтаксис структуры. Элементы тела структуры. Сходства и различия структур и классов. Плюсы и минусы использования структур	4	2
	2 Тип данных enum. Объявление и тонкости использования. Операции с перечислениями	4	2
	Практические занятия	8	3
	1 №61. Создание проектов с использованием перечислений	4	
	2 №62. Создание проектов с использованием структур	4	
		Самостоятельная работа обучающихся Решение задач по образцу по теме «Структуры и перечисления»	2 ⁴
Тема 3.4. Коллекции. Делегаты. События и потоки выполнения	Содержание	32	
	1 Понятие коллекции. Класс List. Возможности, правила работы, встроенные методы. Работа с ArrayList	4	2
	2 Делегаты: понятие, правила описания. Примеры и тонкости использования делегатов	4	
	3 Обработка события. Автоматическое создание обработчиков	4	
	4 Создание библиотеки Использование библиотеки. Примеры	2	
	Практические занятия	16	3
	1 №63. Решение задач на создание событий	4	
	2 №64. Работа с ArrayList	4	
	3 №65. Создание проектов с использованием делегатов	4	
	4 №66. Создание проектов с групповых делегатов.	2	
	5 №67. Создание проектов с использованием обработки события	2	

³ В т.ч. 1 час консультации

⁴ В т.ч. 1 час консультации

	Самостоятельная работа обучающихся Решение задач по образцу по теме «Коллекции. Делегаты. События и потоки выполнения»	2 ⁵	
Итоговая аттестация в форме <i>экзамена</i>		12	
	Всего:	322	

⁵ В т.ч. 1 час консультации

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация программы дисциплины требует наличия лаборатории программирования и баз данных:

стол аудиторный двухместный-12 шт.;

стол преподавателя – 1 шт.;

стулья аудиторные - 24 шт.;

компьютерные столы - 12 шт.;

Автоматизированные рабочие места:

системный блок Dell OptiPlex 3050, Intel Core i5-7500 CPU 3.40GHz, DDR4 8GB, HDD 1TB, Intel HD Graphics 630 1Gb; монитор Dell E2216H 21.5" LED (1920 x 1080);

Microsoft Windows Win10Pro (64) Акт приема-передачи от 31 июля 2017, контракт №0344100007517000020-0008905-01

Программное обеспечение общего и профессионального назначения:

Microsoft Windows Win10Pro (64) Акт приема-передачи от 31 июля 2017, контракт №0344100007517000020-0008905-01;

Microsoft Windows XP Professional Open License: 47818817;

Microsoft Office Professional Plus 2007 Open License:43219389;

7-Zip Свободная лицензия GNU LGPL;

Adobe Acrobat Reader DC Бесплатное программное обеспечение;

Google Chrome Свободная лицензия BSD;

Code::Blocks Свободная лицензия GNU GPLv3;

Eclipse IDE for Java EE Developers Свободная лицензия, NET Framework JDK 8 Свободная лицензия, Microsoft SQL Server Express Edition свободная лицензия, Microsoft Visio Professional,

Microsoft Visual Studio свободная лицензия, MySQL Installer for Windows свободная лицензия, NetBeans свободная лицензия, SQL Server Management Studio свободная лицензия, Microsoft SQL Server Java Connector свободная лицензия, Android Studio свободная лицензия, IntelliJ IDEA свободная лицензия

Сервер GIGANT G-server (Dell PowerEdge R630 /2 CPU x Intel Xeon E5 3.2GHz 8Core/18 x 8Gb DDR4/10 x 1.2TB HS HDD 2.5), Microsoft Windows Server Standard 2016

Маркерная доска – 1, Проектор – 1,

экран – 1

3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

1. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2020. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454452>.

2. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2020. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/456221> ..

3. Нагаева, И.А. Основы алгоритмизации и программирования: практикум : [12+] / И.А. Нагаева, И.А. Кузнецов. — Москва ; Берлин : Директ-Медиа, 2021. — 169 с. : схем. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=598404> (дата обращения: 07.12.2020). — Библиогр.: с. 162-163. — ISBN 978-5-4499-1612-9. — DOI 10.23681/598404. — Текст : электронный.

Дополнительные источники:

1. Кудинов Ю.И. Основы алгоритмизации и программирования [Электронный ресурс]: учебное пособие для СПО/ Кудинов Ю.И., Келина А.Ю.— Электрон. текстовые данные.— Липецк, Саратов: Липецкий государственный технический университет, Профобразование, 2020.— 71 с.— Режим доступа: <http://www.iprbookshop.ru/92834.html>.— ЭБС «IPRbooks»

2. Чурина Т.Г. Основы алгоритмизации и программирования [Электронный ресурс]: учебное пособие для СПО/ Чурина Т.Г., Нестеренко Т.В.— Электрон. текстовые данные.— Саратов, Москва: Профобразование, Ай Пи Ар Медиа, 2020.— 214 с.— Режим доступа: <http://www.iprbookshop.ru/96017.html>.— ЭБС «IPRbooks»

3. Лубашева, Т.В. Основы алгоритмизации и программирования : учебное пособие : [12+] / Т.В. Лубашева, Б.А. Железко. — Минск : РИПО, 2016. — 378 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=463632> (дата обращения: 07.12.2020). — Библиогр. в кн. — ISBN 978-985-503-625-9. — Текст : электронный.

4. Журналы: Программирование

Интернет-ресурсы:

1. Электронный учебник по C++: <http://cgui.ru>

2. Каталог книг по программированию на C, C++: <http://c2p.ru>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
<p><i>Умения:</i></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><i>Знания:</i></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;	<p><i>Оценка выполнения практических работ.</i></p> <p><i>Оценка защиты рефератов, докладов, презентаций.</i></p> <p><i>Оценка выполнения индивидуальных заданий</i></p> <p><i>Оценка ответов на устный опрос</i></p> <p><i>Оценка выполнения тестовых заданий</i></p> <p><i>Экзамен</i></p>

- | | |
|---|--|
| <ul style="list-style-type: none">– 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. | |
|---|--|

