

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ЧПОУ «КОЛЛЕДЖ ГОСУДАРСТВЕННОЙ И МУНИЦИПАЛЬНОЙ СЛУЖБЫ»



УТВЕРЖДАЮ
Директор ЧПОУ «КГИМС»
Хамхоева З.М.

«01» 09 2021г

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
ОПД.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»
для программы подготовки по специальности СПО
09.02.07 «Информационные системы и программирование»

Форма обучения – очная.
Срок обучения - 3 года и 10 месяцев.

Карабулак 2021 г.

Рабочая программа учебной дисциплины разработана в соответствии с требованиями Федерального государственного образовательного стандарта по специальности среднего профессионального образования

09.02.07 Информационные системы и программирование

Организация-разработчик:

ЧПОУ «Колледж государственной и муниципальной службы» Разработчик
Протокол № 3 от «27» августа 2021 г.

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ	13
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ	15
ДИСЦИПЛИНЫ	

**1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ
ОП.4 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

1Л. Область применения рабочей программы

Рабочая программа учебной дисциплины является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО **09.02.07 Информационные системы и программирование**, входящей в укрупненную группу специальностей **09.00.00 Информатика и вычислительная техника**.

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

Учебная дисциплина «Основы алгоритмизации и программирования» принадлежит к профессиональному циклу (П.00) и базируется на знаниях основных математических операций и связи с дисциплиной «Элементы высшей математики».

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

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, инкапсуляции и полиморфизма, наследования и переопределения.

В результате освоения учебной дисциплины у студента должны формироваться общие и профессиональные компетенции, включающие в себя способность:

Код	Наименование общих 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 и иностранном языке.
Код	Наименование профессиональных компетенций
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 1.6.	Разрабатывать модули программного обеспечения для мобильных платформ.
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.
ПК3.1.	Осуществлять ревьюирование программного кода в соответствии с технической документацией.
ПК 3.3.	Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.
ПК 3.4.	Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.

1.4. Количество часов на освоение программы дисциплины:

максимальной учебной нагрузки студента **154** часа, в том числе:
 обязательной аудиторной учебной нагрузки студента **144** часов;
 самостоятельной работы студента **4** часа.

Вид учебной работы	Объем
Суммарная учебная нагрузка во взаимодействии с преподавателем	154
Самостоятельная работа	10
Объем образовательной программы	144
в том числе:	
теоретическое обучение	114
практические занятия	30
самостоятельная работа студента	10
в том числе:	
Контрольная работа по теме «Основные понятия алгоритмизации» (по вас и антам)	1
Решение задач по образцу по теме «Операции и выражения»	1
Решение задач по образцу по теме «Управляющие структуры»	1
Контрольная работа по теме «Структуры данных» (по вариантам)	1
Промежуточная аттестация проводится в форме экзамена в 4 семестре	

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся		Уровень освоения	Объем часов	Осваиваемые элементы компетенций
<i>1</i>	<i>2</i>		<i>3</i>	<i>4</i>	<i>5</i>
Раздел 1. Основы алгоритмизации					
Тема 1.1. Основные понятия алгоритмизации	Содержание учебного материала			6	ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4
	1	Понятие алгоритма, его свойства и виды. Критерии «хорошего» алгоритма. Способы описания алгоритмов: псевдокоды, блок-схема, программа.	1		
	2	Базовые алгоритмические конструкции: линейная, разветвляющаяся, циклическая. Блоксхема: основные элементы, правила составления	2		
	3	Различные комбинации алгоритмических конструкций. Тестовые данные. Математическая модель алгоритма. Классические алгоритмы Евклида Алгоритмы решения нелинейных и линейных уравнений	2		
	Практические занятия			4	
	1	Составление блок-схем линейных алгоритмов, алгоритмов ветвления, циклических алгоритмов. Составление блок-схем алгоритмов усложненной структуры	3		
	2	Составление блок-схем алгоритмов поиска минимального элемента Составление блок-схем алгоритмов бинарного поиска Составление блок-схем алгоритмов сортировки массива простым выбором Составление блок-схем алгоритмов сортировки массива методом обмена	3		
Самостоятельная работа обучающихся контрольная работа по теме «Основные понятия алгоритмизации»				2	
Раздел 2. Основы программирования					ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4
Тема 2.1. Основные элементы процедурного языка	Содержание учебного материала			16	ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4
	1	История языков программирования. Генеалогическое древо C#. Преимущества и недостатки языка.	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.	2	
	4	Базовый синтаксис C#. Пространство имен. Классы. Типы данных: значимые и ссылочные. Преобразование типов: явное и неявное.	2	
	5	Переменные: определение, правила именования. Объявление переменных и их инициализация. Область действия и время существования переменных.	2	
	6	Константы: определение, виды и правила записи в программе. Суффиксы целых и вещественных констант. Escape - последовательности.	2	
	7	Понятие выражения. Математические операторы. Старшинство операторов.	2	
	8	Математические функции (класс Math). Ввод - вывод данных. Операторы присваивания.	2	
	Практические занятия			4
	1	Знакомство со средой Visual.Studio.Net. Создание консольного приложения Организация простейшего ввода-вывода	3	
	2	Разработка линейных программ. Операции и выражения Использование математических функций класса Math	3	
	Самостоятельная работа обучающихся			2
	Решение задач по образцу по теме «Операции и выражения»			
Тема 2.2. Управляющие структуры	Содержание учебного материала			8
	1	Операторы отношения. Оператор if-else: назначение и правила записи. Проверка простых и сложных условий. Вложенные условные операторы.	2	ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4
	2	Использование поразрядных (!, &, , ^) и условных (&&,) логических операторов. Отличие в использовании операторов & и &&, и .	2	
	3	Тернарный оператор: правила использования. Механизм получения случайного числа.	2	
	4	Оператор выбора switch-case: назначение и правила записи. Операторы перехода: break, goto, return.	2	
	Практические занятия			
	1	Создание проектов разветвляющейся структуры Создание проектов с использованием логических операторов Создание проектов с использованием оператора выбора switch-case	3	

Тема 2.3. Операторы цикла	Содержание учебного материала		8	ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4		
	1	Цикл с предусловием (while). Цикл с постусловием (do)			2	
	2	Цикл с параметром (for). Получение таблицы значений			2	
	3	Нахождение суммы и произведения элементов ряда.			2	
	4	Принудительный выход из цикла: операторы break и continue. Бесконечные циклы.	2			
	Практические занятия		4			
1	Создание проектов с использованием циклического оператора while, do, for	3				
2	Создание проектов с использованием break, goto, return	3				
	Самостоятельная работа обучающихся Решение задач по образцу по теме «Управляющие структуры»		4			
Тема 2.4. Структуры данных	Содержание учебного материала		16	ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4		
	1	Массивы: определение, виды. Объявление одномерного массива. Варианты инициализации. Ввод и вывод одномерных массивов.			2	
	2	Стандартные операции для работы с массивами: заполнение случайными значениями и значениями по формуле			2	
	3	Нахождение суммы и произведения. Нахождение максимума (минимума)			2	
	4	Подсчет количества элементов, удовлетворяющих определенному условию.			2	
	5	Обработка одномерных массивов: сортировка, поиск элементов			2	
	6	Цикл foreach.			2	
	7	Двумерные массивы: объявление, ввод и вывод.			2	
	8	Работа с двумерными массивами по строкам и по столбцам			2	
	Практические занятия				6	
	1	Создание проектов с использованием одномерных и двумерных массивов Создание проектов с использованием методов сортировки и поиска				3
	2	Создание проектов с использованием цикла foreach. Создание проектов с использованием методов класса Array: Sort, Indexof, BinarySearch				3
	3	Создание проектов с использованием ступенчатых массивов, массивов символов Создание проектов с использованием строк			3	
		Самостоятельная работа обучающихся Контрольная работа по теме «Структуры данных» (по вариантам)			2	

Тема 2.5 Потоки. Файлы	Содержание учебного материала		8	ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4	
	1	Понятие потока. Механизм буферизации. Классы библиотеки .NET для работы с потоками. Виды доступа к файлам. Объект FileStream.			2
	2	Классы StreamWriter и StreamReader.			2
	3	Основные операции при работе с файлами. Способы работы с текстовыми файлами.			
	4	Обработка текстовых файлов. Работа с каталогами и файлами			2

Практические занятия			4
1	Создание проектов с использованием class Random и обработки исключительных ситуаций	2	
2	Создание проектов с использованием операторов checked и unchecked. Создание проектов с использованием многопоточных приложений	2	

Раздел 3.
Объектноориентированная модель программирования

Тема 3.1 Основные принципы объектно-ориентированного программирования	Содержание учебного материала		17	ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4	
	1	Понятие класса и объекта. Характеристики объекта: поля, свойства, методы, события.			2
	2	Основные принципы объектно-ориентированного программирования: наследование, полиморфизм, инкапсуляция.			2
	3	Общая форма определения класса. Модификаторы доступа к элементам класса: public, private, protected, internal. Примеры создания классов.			2
	4	Переменные ссылочного типа и присваивание. Побочные эффекты множественных ссылок.			2
	5	Метод: понятие, правила записи. Вызов метода. Передача параметров по значению. Правило триединого соответствия параметров и аргументов: по количеству, типам и по порядку следования.			2
	6	Создание методов, возвращающих значения. Способы размещения методов. Конструкторы.			2
	7	Инкапсуляция как управление доступом к данным. Свойства класса: понятие, виды, правила записи. Решение задач на создание классов со свойствами.			2

8	Наследование и полиморфизм. Иерархия классов: понятие, преимущества. Синтаксис наследования.	2
9	Скрытие и перекрытие методов. Ключевые слова virtual, override. Вызов методов базового класса («родителя»): ключевое слово base. Тонкости использования конструкторов в иерархически связанных между собой классах.	2
Практические занятия		16

1	Создание проектов с использованием текстовых элементов управления и окон сообщений. Создание проектов с использованием диалоговых окон	2
2	Создание проектов с использованием кнопок Создание проектов с использованием компонентов для создания меню	2
3	Создание проектов с использованием графических объектов Создание проектов с использованием многооконного интерфейса	2
4	Создание проектов с использованием списков Создание проектов с использованием переключателей	2
5	Создание проектов с использованием текстовых редакторов Создание проектов с использованием перегрузки методов	2
6	Создание проектов с использованием рекурсивных методов Создание проектов с использованием метода Main	2
7	Создание проектов с использованием методов с переменным количеством аргументов Создание классов, иерархически связанных между собой	2
8	Создание проектов с использованием абстрактных классов Создание проектов с использованием виртуальных методов, наследования	2
Тема 3.2 Интерфейсы	Содержание учебного материала	5
1	Интерфейсы: назначение, правила написания. Способы реализации интерфейсов.	1
2	Работа с объектами через интерфейсы.	1
3	Операторы is и as. Восходящее и нисходящее приведение. Наследование в интерфейсах.	1
4	Сходства и различия интерфейсов, абстрактных классов и обычных классов.	1
5	Стандартные интерфейсы .NET: IComparable, ICloneable, IEnumerable	1
Практические занятия		4
1	Создание проектов с использованием интерфейса IComparable, ICloneable, IEnumerable	2

ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4

	2	Создание проектов с использованием операторов is и as. Создание проектов с использованием наследования в интерфейсах	2		
Тема 3.3 Структуры и перечисления	Содержание учебного материала			6	ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4
	1	Назначение и синтаксис структуры. Элементы тела структуры.	2		
	2	Сходства и различия структур и классов. Плюсы и минусы использования структур	2		
	3	Тип данных епшп: объявление и тонкости использования. Операции с перечислениями.	2		
	Практические занятия			2	
	1	Создание проектов с использованием перечислений. Создание проектов с использованием структур	2		
Тема 3.4. Коллекции. Делегаты. События и потоки выполнения	Содержание учебного материала			4	ОК 1, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1- ПК 1.6, ПК 2.4, 2.5, ПК 3.1, 3.3, 3.4
	1	Понятие коллекции. Класс List: возможности, правила работы, встроенные методы. Работа с ArrayList.	2		
	2	Делегаты: понятие, правила описания. Примеры и тонкости использования делегатов. Обработка события: автоматическое создание обработчиков. Создание библиотеки. Использование библиотеки	2		
	Практические занятия			2	
	1	Решение задач на создание событий. Работа с ArrayList. Создание проектов с использованием делегатов. Создание проектов с использованием групповых делегатов	3		
Всего:				144	

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация программы дисциплины требует наличия следующих лабораторий:

1 Лаборатория «Программирования и баз данных»

- Автоматизированные рабочие места на 12-15 обучающихся (Процессор не ниже Core i3, оперативная память объемом не менее 8 Гб);
- Автоматизированное рабочее место преподавателя (Процессор не ниже Core i3, оперативная память объемом не менее 8 Гб);
- Сервер в лаборатории (8-х ядерный процессор с частотой не менее 3 ГГц, оперативная память объемом не менее 16 Гб, жесткие диски общим объемом не менее 1 Тб, программное обеспечение: WindowsServer 2012 или более новая) или выделение аналогичного по характеристикам виртуального сервера из общей фермы серверов
- Проектор и экран;
- Маркерная доска;
- Программное обеспечение общего и профессионального назначения, в том числе включающее в себя следующее ПО:
Eclipse IDE for Java EE Developers, .NET Framework JDK 8, Microsoft SQL Server Express Edition, Microsoft Visio Professional, Microsoft Visual Studio, MySQL Installer for Windows, NetBeans, SQL Server Management Studio, Microsoft SQL Server Java Connector, Android Studio, IntelliJ IDEA.

3.2. Информационное обеспечение обучения

Перечень используемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники (печатные издания):

1. Пахомов Б. И. *С# начинающих*. Издательство: БХВ-Петербург Год: 2015 2. Эндрю Стиллмен, Дженнифер Грин Изучаем С#. Третье издание Издательство Design Patterns via С#.

Дополнительные источники (печатные издания)

1. Биллиг В. А. Основы программирования на **С#**. - М.: Изд-во «Интернет-университет информационных технологий - ИНТУИТ . ру», 2006. - 488 с.
2. Вирт Н. Алгоритмы и структуры данных. - СПб: Невский диалект, 2001. - 352 с.
3. Гуннерсон Э. Введение в **С#**. Библиотека программиста. - СПб.: Питер, 2001. - 304 с.
4. Майо Дж. *С# Builder*. Быстрый старт. — М.: Бином, 2005. - 384 с.
5. Петцольд Ч, Программирование для MS Windows на С#. Т. 1. — М.: Изда-тельско-торговый дом «Русская Редакция», 2002. — 576 с.
6. Пономарев В. А. Программирование на C++ / C# в Visual Studio .NET 2003. Серия «Мастер программ». — СПб.: БХВ-Петербург, 2004. — 352 с.
7. Шилдт Г. *С#*: Учебный курс. - СПб.: Питер, 2002. - 512 с: ил.
8. Microsoft Corporation. Разработка Windows-приложений на Microsoft Visual Basic .NET и Microsoft Visual С# .NET. Учебный курс. Сертификационный экзамен № 70-306, 70-316. — М.: Издательско-торговый дом «Русская Редакция», 2003. - 512 с.
9. Саша Годштейн, Дима Зурбалева, Идо Флатов Оптимизация приложений на платформе .NET Издательство: ДМК Пресс Год: 2014 **Интернет-ресурсы**
1. *С# начинающих*. Автор: Пахомов Б. И. Издательство: БХВ-Петербург Год: 2014 Язык: Русский ISBN: 978-5-9775-0943-5 Страниц: 432
Источник: <http://forcoder.ru/c-sharp/pages/2>
2. Изучаем С#. Третье издание Автор: Эндрю Стиллмен, Дженнифер Грин Издательство: 978-5-49600867-9 Год: 2014 Язык: Русский ISBN: 978-5-496-00867-9 Страниц: 816 Формат: pdf Размер: 23 Мб Источник: <http://forcoder.ru/c-sharp/pages/2>
3. Design Patterns via С#. Приемы объектно-ориентированного проектирования Источник: <http://forcoder.ru/c-sharp/pages/2> Издательство: ИТ\ГЖ Год: 2015 Язык: Русский

Страниц: 288 Формат: pdf Размер: 13 Мб Источник:

<http://forcoder.ru/c-sharp/pages/2> 4. Оптимизация

приложений на платформе .NET

Саша Годштейн, Дима Зурбалеv, Идо Флатов Издательство: ДМК Пресс Год: 2014 Язык: Русский

ISBN: 978-5-94074-944-8 Страниц: 522 Формат: pdf Источник: <http://forcoder.ru/c-sharp/pages/3>

3.3. Организация образовательного процесса

Дисциплина ОП.4 Основы алгоритмизации и программирования изучается в течение 4 семестра. Форма проведения консультаций для обучающихся- индивидуальная.

3.4. Кадровое обеспечение образовательного процесса

Реализация образовательной программы обеспечивается педагогическими работниками образовательной организации, а также лицами, привлекаемыми к реализации образовательной программы на условиях гражданско-правового договора, в том числе из числа руководителей и работников организаций, направление деятельности которых соответствует области профессиональной деятельности, имеющих стаж работы в данной профессиональной области не менее 3 лет.

Квалификация педагогических работников образовательной организации должна отвечать квалификационным требованиям, указанным в квалификационных справочниках и профессиональных стандартах.

Педагогические работники, привлекаемые к реализации образовательной программы, должны получать дополнительное профессиональное образование по программам повышения квалификации, в том числе стажировок в организациях направление деятельности, которых соответствует области профессиональной деятельности, не реже 1 раза в 3 года с учетом расширения спектра профессиональных компетенций.

Результаты обучения (освоенные умения, усвоенные знания)	Основные показатели оценки результата	Формы и методы контроля и оценки
Перечень знаний, осваиваемых в рамках дисциплины:		
Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.	90-100 % правильных ответов - «5»; 70-89% правильных ответов - «4»; 50-69 % правильных ответов - «3»; менее 50 % - «2»	-устный опрос, тестирование выполнение индивидуальных заданий различной сложности
Эволюция языков программирования, их классификация, понятие системы программирования	90-100 % правильных ответов - «5»; 70-89% правильных ответов - «4»; 50-69 % правильных ответов - «3»; менее 50 % - «2»	оценка ответов в ходе эвристической беседы, тестирование
Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.	90-100 % правильных ответов - «5»; 70-89% правильных ответов - «4»; 50-69 % правильных ответов - «3»; менее 50 % - «2»	-оценка ответов в ходе эвристической беседы выполнение индивидуальных заданий различной сложности - письменная проверка в виде контрольной работы;
Подпрограммы, составление библиотек подпрограмм.	90-100 % правильных ответов - «5»; 70-89% правильных ответов - «4»; 50-69 % правильных ответов - «3»; менее 50 % - «2»	-оценка ответов в ходе эвристической беседы,
Объектноориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма.	90-100 % правильных ответов - «5»; 70-89% правильных ответов - «4»; 50-69 % правильных ответов - «3»; менее 50 % - «2»	-оценка ответов в ходе эвристической беседы выполнение индивидуальных заданий различной сложности - письменная проверка в виде контрольной работы;
Перечень умений, осваиваемых в рамках дисциплины:		
Разрабатывать алгоритмы для конкретных задач	90-100 % правильных ответов и выполненных действий - «5»; 70- 89% правильных ответов и выполненных действий - «4»; 50-69 % правильных ответов и выполненных действий - «3»; менее 50 % - «2»	-выполнение индивидуальных заданий различной сложности оценка результатов выполнения практических работ

Использовать программы для графического отображения алгоритмов	90-100 % правильных ответов и выполненных действий - «5»; 70- 89% правильных ответов и выполненных действий - «4»; 50-69 % правильных ответов и выполненных действий - «3»; менее 50 % - «2»	-оценка результатов выполнения практических работ - выполнение индивидуальных заданий различной сложности
Определять сложность работы алгоритмов.	90-100 % правильных ответов и выполненных действий - «5»; 70- 89% правильных ответов и выполненных действий - «4»; 50-69 % правильных ответов и выполненных действий - «3»; менее 50 % - «2»	-устный опрос -тестирование -оценка результатов выполнения практических работ
Работать в среде программирования.	90-100 % правильных ответов и выполненных действий - «5»; 70- 89% правильных ответов и выполненных действий - «4»; 50-69 % правильных ответов и выполненных действий - «3»; менее 50 % - «2»	-устный опрос, тестирование, -оценка результатов выполнения практических работ - выполнение индивидуальных заданий различной
Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.	90-100 % правильных ответов и выполненных действий - «5»; 70- 89% правильных ответов и выполненных действий - «4»; 50-69 % правильных ответов и выполненных действий - «3»; менее 50 % - «2»	сложности устный опрос, тестирование, -оценка результатов выполнения практических работ - выполнение индивидуальных заданий различной сложности
Оформлять код программы в соответствии со стандартом кодирования	90-100 % правильных ответов и выполненных действий - «5»; 70- 89% правильных ответов и выполненных действий - «4»; 50-69 % правильных ответов и выполненных действий - «3»; менее 50 % - «2»	-оценка результатов выполнения практических работ
Выполнять проверку, отладку кода программы	90-100 % правильных ответов и выполненных действий - «5»; 70- 89% правильных ответов и выполненных действий - «4»; 50-69 % правильных ответов и выполненных действий - «3»; менее 50 % - «2»	-оценка результатов выполнения практических работ