ГОСУДАРСТВЕННОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ТУЛЬСКОЙ ОБЛАСТИ

«ТУЛЬСКИЙ ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ»

**УТВЕРЖДАЮ**

**Директор ГПОУ ТО «ТЭК»**

**\_\_\_\_\_\_\_\_\_\_\_ А.В. Макарова**

**Приказ № 25**

**«30» апреля 2025 года**

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

Щекино

2025 год

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

Организация-разработчик: **Государственное профессиональное образовательное учреждение Тульской области «Тульский экономический колледж»**

Разработчики:

Каргина Ольга Ивановна, преподаватель высшей квалификационной категории государственного профессионального образовательного учреждения Тульской области «Тульский экономический колледж»

Рабочая программа рекомендована предметно-цикловой комиссией № 3 Государственного профессионального образовательного учреждения Тульской области «Тульский экономический колледж»

Протокол № 8 от «29» апреля 2025 года

Председатель ПЦК № 3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.И. Каргина

Согласовано:

Заместитель директора по учебной работе \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.В. Кошелева

«30» апреля 2025 года

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
|  | стр. |
| 1. **ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ** | 4 |
| 1. **СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ** | 5 |
| 1. **УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ** | 11 |
| 1. **КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ** | 12 |
| 1. **ПРИЛОЖЕНИЕ 1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ УЧЕБНОЙ ДИСЦИПЛИНЫ** | 15 |

**1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**ОП.4 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

**1.1. Цель и место дисциплины в структуре образовательной программы**

Цель дисциплины «ОП.4 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»: формирование у обучающихся знаний основ алгоритмизации, а также умений и базовых навыков программирования на объектно-ориентированном языке C#.

Дисциплина «ОП.4 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» включена в обязательную часть профессионального цикла образовательной программы.

1.2. Планируемые результаты освоения дисциплины

Результаты освоения дисциплины соотносятся с планируемыми результатами освоения образовательной программы, представленными в матрице компетенций выпускника.

В результате освоения дисциплины обучающийся должен:

|  |  |  |  |
| --- | --- | --- | --- |
| **Код ОК,**  **ПК** | **Уметь** | **Знать** | **Владеть навыками** |
| ОК.01 | распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части; определять этапы решения задачи, составлять план действия, реализовывать составленный план, определять необходимые ресурсы; | основные источники информации и ресурсы для решения задач и/или проблем в профессиональном и/или социальном контексте; | *-* |
| ОК.02 | определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации;  выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска; | Номенклатуру информационных источников, применяемых в профессиональной деятельности;  формат оформления результатов поиска информации | *-* |
| ОК.04 | организовывать работу коллектива и команды;  взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности | психологические основы деятельности коллектива;  психологические особенности личности | - |
| ОК.09 | понимать тексты на базовые профессиональные темы;  участвовать в диалогах на знакомые общие и профессиональные темы;  строить простые высказывания о себе и о своей профессиональной деятельности;  кратко обосновывать и объяснять свои действия (текущие и планируемые):  писать простые связные сообщения на знакомые или интересующие профессиональные темы | лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности  особенности произношения  правила чтения текстов профессиональной направленности. | - |
| ПК 2.5. | выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace); создавать классы- исключения на основе базовых классов; выявлять ошибки в системных компонентах на основе спецификаций | основы организации инспектирования и верификации; встроенные и основные специализированные инструменты анализа качества программных продуктов | инспектирование разработанных программных модулей на предмет соответствия стандартам кодирования. |

**1.3. Обоснование часов вариативной части ОПОП-П**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Дополнительные знания, умения** | **№, наименование темы** | **Объем часов** | **Обоснование включения в рабочую программу** |
| 1 | Использовать знания о способах описания алгоритма и основных элементах блок-схем; анализировать потребность системы в ресурсах; применять умения построения базовых алгоритмических конструкций и алгоритмов сложной структуры | Тема 1.1. Основные понятия алгоритмизации | 8 | Для расширения перечня осваиваемых умений, участия в соревнованиях в рамках Регионального чемпионата «Профессионалы» |
| 2 | Использовать знания интегрированной среды разработки Visual Studio .NET; производить создание консольного приложения, линейных программ; прописывать пользовательские функции для системы | Тема 2.1. Основные элементы процедурного языка | 12 | Для расширения перечня осваиваемых умений, участия в соревнованиях в рамках Регионального чемпионата «Профессионалы» |
| 3 | Использовать знания о работе с массивами; создавать проекты с использованием одномерных, двумерных и многомерных массивов | Тема 2.4.  Структуры  данных | 14 | Для расширения перечня осваиваемых умений, участия в соревнованиях в рамках Регионального чемпионата «Профессионалы» |
| 4 | Использовать знания принципов объектно-ориентированного программирования и применять умения в работе; анализировать потребность системы в таких ресурсах | Тема 3.1.  Основные принципы  объектно-  ориентированного  программирования | 18 | Для расширения перечня осваиваемых умений, участия в соревнованиях в рамках Регионального чемпионата «Профессионалы» |
|  | ИТОГО |  | 52 |  |

**2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**2.1. Трудоемкость освоения дисциплины**

|  |  |  |
| --- | --- | --- |
| **Наименование составных частей дисциплины** | **Объем в часах** | **В т.ч. в форме практ. подготовки** |
| Учебные занятия | 174 | 80 |
| *Курсовая работа (проект)* | - | - |
| Самостоятельная работа | 10 | - |
| Промежуточная аттестация *(экзамен)* | 6 | - |
| Всего | **190** | **80** |

2.2. Содержание дисциплины

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование разделов и тем** | **Содержание учебного материала и формы организации деятельности обучающихся** | **Объем, акад. ч / в том числе  в форме практической подготовки, акад. ч** | **Коды компетенций,** **формированию которых способствует элемент программы** |
| ***1*** | ***2*** | ***3*** | ***4*** |
| **Раздел 1. Основы алгоритмизации** | | **20 / 8** |  |
| **Тема 1.1.**  **Основные понятия**  **алгоритмизации** | **Содержание** | ***10*** | ОК 01, ОК 02, ОК 04, ОК 09 |
| 1.Понятие алгоритма, его свойства и виды. Способы описания алгоритмов: псевдокоды. Блок-схема: основные элементы, правила составления |
| 2.Базовые алгоритмические конструкции: линейная, разветвляющаяся, циклическая. Алгоритмы сложной структуры. |
| 3.Классические алгоритмы Евклида |
| 4.Алгоритмы поиска. Последовательный поиск. Бинарный поиск. |
| 5.Алгоритмы сортировки: пузырьковая сортировка, сортировка слиянием, быстрая сортировка, сортировка прямым выбором. |
| **Практические занятия и лабораторные работы** | ***8*** | ОК 01, ОК 02, ОК 04, ОК 09 |
| 1.Составление блок-схем линейных алгоритмов | *2* |
| 2.Составление блок-схем алгоритмов ветвления | *2* |
| 3.Составление блок-схем циклических алгоритмов | *2* |
| 4.Составление блок-схем алгоритмов усложненной структуры | *2* |
| **Самостоятельная работа обучающихся**  контрольная работа по теме «Основные понятия алгоритмизации» | ***2*** | ОК 01, ОК 02, ОК 04, ОК 09 |
| **Раздел 2. Основы программирования** | | **122/52** |  |
| **Тема 2.1.**  **Основные**  **элементы**  **процедурного языка** | **Содержание** | ***20*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Платформа .NET Framework: назначение, составные элементы, тонкости компиляции. |
| 2.Знакомство со средой программирования. Обзор интегрированной среды разработки Visual Studio .NET. |
| 3.Создание, сохранение и закрытие проектов и решений. Структура программы. Точка входа. |
| 4.Базовый синтаксис С#. Пространство имен. Классы. |
| 5.Типы данных: значимые и ссылочные. Преобразование типов: явное и неявное. |
| 6.Переменные: определение, правила именования. Объявление переменных и их инициализация. Область действия и время существования переменных. |
| 7.Константы: определение, виды и правила записи в программе. Суффиксы целых и вещественных констант. Escape – последовательности. |
| 8.Понятие выражения. Математические операторы. Старшинство операторов. |
| 9.Математические функции (класс Math). |
| 10.Ввод – вывод данных. Операторы присваивания. |
| **Практические занятия и лабораторные работы** | ***10*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Знакомство со средой Visual.Studio.Net | *2* |
| 2.Создание консольного приложения | *2* |
| 3.Организация простейшего ввода-вывода | *2* |
| 4.Разработка линейных программ. Операции и выражения | *2* |
| 5.Использование математических функций класса Math | *2* |
| **Самостоятельная работа обучающихся**  1.Решение задач по образцу по теме «Операции и выражения» | ***2*** |  |
| **Тема 2.2.**  **Управляющие**  **структуры** | **Содержание** | ***12*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Операторы отношения. Оператор if-else: назначение и правила записи. Проверка простых и сложных условий. Вложенные условные операторы. |
| 2.Использование поразрядных (!,&, |, ^) и условных (&&, ||) логических операторов. Отличие в использовании операторов & и &&, | и ||. |
| 3.Тернарный оператор: правила использования. |
| 4.Механизм получения случайного числа. |
| 5.Оператор выбора switch-case: назначение и правила записи. |
| 6.Операторы перехода: break, goto, return. |
| **Практические занятия и лабораторные работы** | ***6*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Создание проектов разветвляющейся структуры | *2* |
| 2.Создание проектов с использованием логических операторов | *2* |
| 3.Создание проектов с использованием оператора выбора switch-case | *2* |
| **Тема 2.3.**  **Операторы**  **цикла** | **Содержание** | ***12*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Цикл с предусловием (while) |
| 2.Цикл с постусловием (do) |
| 3.Цикл с параметром (for). |
| 4.Получение таблицы значений |
| 5.Нахождение суммы и произведения элементов ряда. |
| 6.Принудительный выход из цикла: операторы break и continue. Бесконечные циклы. |
| **Практические занятия и лабораторные работы** | ***10*** | ОК 01, ОК 02, ОК 04, ОК 09  ПК 2.5 |
| 1. Создание проектов с использованием циклического оператора while | *2* |
| 2. Создание проектов с использованием циклического оператора do | *2* |
| 3. Создание проектов с использованием циклического оператора for | *2* |
| 4. Создание проектов с использованием break, goto, return | *2* |
| 5. Решение задач с применением циклов | *2* |
| **Тема 2.4.**  **Структуры**  **данных** | **Содержание** | ***14*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Массивы: определение, виды. Объявление одномерного массива. Варианты инициализации. |
| 2.Ввод и вывод одномерных массивов. |
| 3.Стандартные операции для работы с массивами: заполнение случайными значениями и значениями по формуле |
| 4.Нахождение суммы и произведения. Нахождение максимума (минимума) |
| 5.Подсчет количества элементов, удовлетворяющих определенному условию. |
| 6.Обработка одномерных массивов: сортировка, поиск элементов |
| 7.Цикл foreach. Двумерные массивы: объявление, ввод и вывод. |
| **Практические занятия и лабораторные работы** | ***12*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1. Создание проектов с использованием одномерных массивов | *2* |
| 2. Создание проектов с использованием двумерных массивов | *2* |
| 3. Создание проектов с использованием методов сортировки | *2* |
| 4. Создание проектов с использованием методов поиска | *2* |
| 5. Создание проектов с использованием цикла foreach | *2* |
| 6. Создание проектов с использованием методов класса Array: Sort, Indexof, BinarySearch | *2* |
| **Самостоятельная работа обучающихся**  Решение задач по образцу по теме «Структуры данных» (по вариантам) | ***2*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| **Тема 2.5**  **Потоки. Файл** | **Содержание** | ***8*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Понятие потока. Механизм буферизации. Классы библиотеки .NET для работы с потоками |
| 2.Виды доступа к файлам. Объект FileStream. |
| 3.Классы StreamWriter и SreamReader. |
| 4. Основные операции при работе с файлами. Способы работы с текстовыми файлами. |
| **Практические занятия и лабораторные работы** | ***14*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Создание проектов с использованием class Random | *2* |
| 2.Создание проектов с использованием обработки исключительных ситуаций | *2* |
| 3.Создание проектов с использованием операторов checked и unchecked | *2* |
| 4.Создание проектов с использованием многопоточных приложений | *2* |
| 5.Создание проектов с использованием текстовых файлов | *2* |
| 6.Создание проектов с использованием двоичных файлов | *2* |
| 7.Создание проектов с использованием асинхронного ввода-вывода | *2* |
| **Раздел 3. Объектно- ориентированная модель программирования** | | **42/20** |  |
| **Тема 3.1**  **Основные принципы**  **объектно-**  **ориентированного**  **программирования** | **Содержание** | ***110*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Понятие класса и объекта. Характеристики объекта: поля, свойства, методы, события. Основные принципы объектно-ориентированного программирования: наследование, полиморфизм, инкапсуляция. |
| 2. Общая форма определения класса. Модификаторы доступа к элементам класса: public, private, protected, internal. Примеры создания классов. |
| 3. Переменные ссылочного типа и присваивание. Побочные эффекты  множественных ссылок. |
| 4. Метод: понятие, правила записи. Вызов метода. Передача параметров по значению. Правило триединого соответствия параметров и аргументов: по количеству, типам и по порядку следования. |
| 5. Создание методов, возвращающих значения. Способы размещения методов. Конструкторы. |
| **Практические занятия и лабораторные работы** | ***18*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Решение задач на создание классов со свойствами. | *2* |
| 2.Создание проектов с использованием текстовых элементов управления и окон сообщений | *2* |
| 3.Создание проектов с использованием диалоговых окон | *2* |
| 4.Создание проектов с использованием кнопок | *2* |
| 5.Создание проектов с использованием компонентов для создания меню | *2* |
| 6.Создание проектов с использованием графических объектов | *2* |
| 7.Создание проектов с использованием многооконного интерфейса | *2* |
| 8.Создание проектов с использованием списков | *2* |
| 9.Создание проектов с использованием переключателей | *2* |
| **Самостоятельная работа обучающихся**  Решение задач по образцу по теме «Визуальные компоненты | ***2*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| **Тема 3.2**  **Интерфейсы** | **Содержание** | ***8*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Интерфейсы: назначение, правила написания. Способы реализации интерфейсов. Работа с объектами через интерфейсы. |
| 2. Операторы is и as. Восходящее и нисходящее приведение. Наследование в интерфейсах. |
| 3. Сходства и различия интерфейсов, абстрактных классов и обычных классов. |
| 4. Стандартные интерфейсы .NET: IComparable, IClonable, IEnumerable |
| **Практические занятия и лабораторные работы** | ***2*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| 1.Создание проектов с использованием общедоступных интерфейсов | *2* |
| **Самостоятельная работа обучающихся**  Подготовка конспекта по теме «Способы реализации интерфейсов» | ***2*** | ОК 01, ОК 02, ОК 04, ОК 09 ПК 2.5 |
| **Промежуточная аттестация** | | ***6*** |  |
| Всего: | | **190 / 80** |  |

3. Условия реализации ДИСЦИПЛИНЫ

3.1. Материально-техническое обеспечение

Лаборатория *«*Программирования и баз данных»*,* оснащённая в соответствии с приложением 3 ОПОП-П.

3.2. Учебно-методическое обеспечение

**3.2.1. Основные печатные и/или электронные издания**

1. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебник для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2025. — 322 с.

2. Федорова, Г.Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие / Г. Н. Федорова. – М.: КУРС: ИНФРА-М, 2025. – 336 с.

3. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 196 с.

***3.2.2. Дополнительные источники***

* + - 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.Дрейер М. C# для школьников: Учебное пособие / М. Дрейер. Перевод с англ. под ред. В. Биллига - М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2009. - 128 с.

5.Петцолъд Ч. Программирование для MS Windows на С#. Т. 1. — М.: Издательско-торговый дом «Русская Редакция», 2002. — 576 с.

6.Пономарев В. А. Программирование на С + + / С # 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 C# .NET. Учебный курс. Сертификационный экзамен № 70-306, 70-316. — М.: Издательско-торговый дом «Русская Редакция», 2003. - 512 с.

9.Пахомов Б.И. С# для начинающих. Издательство: БХВ-Петербург, 2014. - 432 с.

10.Стиллмен Э., Грин Дж. Изучаем C#. 3-е изд. - СПб.: Питер, 2014. - 816 с.

**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: 978-5-496-00867-9 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 C#. Приемы объектно-ориентированного проектирования

Источник: <http://forcoder.ru/c-sharp/pages/2> Издательство: ITVDN 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 Зурбалев, 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>

**4.КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Результаты обучения*** | ***Критерии оценки*** | | ***Методы оценки*** | |
| ***Перечень знаний, осваиваемых в рамках дисциплины:*** | | | | | |
| Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. | «Отлично» - теоретическое содержание курса освоено полностью, без пробелов, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.  «Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.  «Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.  «Неудовлетворительно» - теоретическое содержание курса не освоено, выполненные учебные задания содержат грубые ошибки. | | устный опрос, тестирование, выполнение индивидуальных заданий различной сложности контрольные вопросы по теме;  оценка ответов в ходе эвристической беседы,  подготовка презентаций;  устный опрос, тестирование, выполнение индивидуальных заданий различной сложности письменная проверка в виде диктанта; собеседование | |
| Эволюция языков программирования, их классификация, 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. |
| основные источники информации и ресурсы для решения задач и/или проблем в профессиональном и/или социальном контексте; |
| Номенклатуру информационных источников, применяемых в профессиональной деятельности;  формат оформления результатов поиска информации |
| психологические основы деятельности коллектива;  психологические особенности личности |
| лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности  особенности произношения  правила чтения текстов профессиональной направленности. |
| основы организации инспектирования и верификации; встроенные и основные специализированные инструменты анализа качества программных продуктов |
| ***Перечень умений, осваиваемых в рамках дисциплины:*** | | | | | |
| Разрабатывать алгоритмы для конкретных задач | | «Отлично» - умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.  «Хорошо» - некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.  «Удовлетворительно» - необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.  «Неудовлетворительно» - необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки. | | оценка ответов в ходе эвристической беседы,  подготовка презентаций домашние задания проблемного характера;  практические задания по работе с информацией, документами, литературой; подготовка и защита индивидуальных и групповых заданий |
| 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 | |
| распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части; определять этапы решения задачи, составлять план действия, реализовывать составленный план, определять необходимые ресурсы; | |
| определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации;  выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска; | |
| организовывать работу коллектива и команды;  взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности | |
| понимать тексты на базовые профессиональные темы;  участвовать в диалогах на знакомые общие и профессиональные темы;  строить простые высказывания о себе и о своей профессиональной деятельности;  кратко обосновывать и объяснять свои действия (текущие и планируемые):  писать простые связные сообщения на знакомые или интересующие профессиональные темы | |
| выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace); создавать классы- исключения на основе базовых классов; выявлять ошибки в системных компонентах на основе спецификаций | |

**ПРИЛОЖЕНИЕ 1.**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**1. Общие положения**

**Фонд оценочных средств (ФОС)** предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины Основы алгоритмизации и программирования.

ФОС включают контрольные материалы для промежуточной аттестации в форме экзамена (3 семестр) и дифференцированного зачета (4 семестр).

**2. Результаты освоения дисциплины, подлежащие проверке**

|  |  |
| --- | --- |
| ***КОД*** | **Освоенные умения, усвоенные знания** |
| З1 | Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. |
| З2 | Эволюция языков программирования, их классификация, понятие системы программирования |
| З3 | Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. |
| З4 | Подпрограммы, составление библиотек подпрограмм. |
| З5 | 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. |
| У1 | Разрабатывать алгоритмы для конкретных задач |
| У2 | Использовать программы для графического отображения алгоритмов |
| У3 | Определять сложность работы алгоритмов |
| У4 | Работать в среде программирования. |
| У5 | Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. |
| У6 | Оформлять код программы в соответствии со стандартом кодирования |
| У7 | Выполнять проверку, отладку кода программы |

**3. Структура контрольного задания**

**3.1. Текст типового задания**

**3. Структура контрольного задания**

**3.1. Экзамен (4 семестр)**

**Текст типового задания**

**Задание 1:**

Проверяемые результаты обучения: З1, З2, З3, З4.

Обучающимся предлагается ответить на два теоретических вопроса по языку программирования C# из перечня:

1. Средой разработки приложений Visual Studio.NET называется ...

*Совокупность средств, с помощью которых программисты пишут, корректируют, преобразуют в машинные коды, отлаживают и запускают программы*

1. Совокупность средств, с помощью которых программисты пишут, корректируют, преобразуют в машинные коды, отлаживают и запускают программы, называется … *Visual Studio.NET*
2. Совокупность средств, с помощью которых программисты решают задачи межъязыкового взаимодействие, возможность применять одновременно несколько языков программирования, относится к …

*Visual Studio.NET*

1. Совокупность средств, с помощью которых программисты решают задачи переносимости программ на различные типы компьютеров, относится к …

*.NET*

1. Что содержит файл сборки? *Код на языке CIL и метаданные*
2. Файл с каким расширением содержит сборка? *EXE DLL*
3. Файл с каким расширением содержит проект? *Csproj*
4. Файл с каким расширением содержит решения? *Sln*
5. Файл с каким расширением содержит исходный код программы? *Cs*
6. Что содержит файл с расширением .cs? *исходный код программы*
7. Что содержит файл с расширением .csproj? *содержит проект*
8. Что содержит файл с расширением .sln? *содержит решения*
9. Что содержит файл с расширением .exe? *содержит сборку*
10. Что содержит файл с расширением .dll? *содержит сборку*
11. Как обеспечивается переносимость программ различных языков программирования в платформе **.NET?** *Cil – промежуточный язык*
12. За счет чего возможна переносимость программ различных языков программирования в платформе .NET? *промежуточный язык*
13. Что содержит .NET Framework платформы .NET? *Библиотека классов*
14. Что содержит CTS платформы .NET? *Система типов*
15. Что содержит CLR платформы .NET? *Среда выполнения*
16. Что содержит CIL платформы .NET? *Промежуточный язык*
17. Что такое ООП? *Объектно-ориентированное программирование*
18. Для чего необходимо пространство имен программы?

*Обьединение взаимно связанных классов*

1. Переменной S необходимо присвоить значение ''привет''.Каким образом данная переменная должна быть объявлена в программе? *String*
2. Переменной S присваивается следующее значение S = 18.34. Каким образом данная переменная должна быть объявлена в программе? *Double*
3. Переменной S присваивается следующее значение S = ‘7’. Каким образом данная переменная должна быть объявлена в программе? *CHAR*
4. Переменной S присваивается следующее значение S = False. Каким образом данная переменная должна быть объявлена в программе? *bool*
5. Что означает следующая запись: double A, B; ? *A, B- дробные*
6. Для чего используются два подряд символа ‘/’ – ‘//’ ? *Комментарии*
7. Для чего используется служебное слово using ? *Подключение пространства имен*
8. Для чего используются символы ‘/\*’ . . . . ‘\*/’ ? *Комментарии*
9. Для чего используются символы ‘{’ . . . . ‘}’ ? *Тело*
10. Что означает слово const в записи const char CIMV = 'y';? *Обозначает константу*
11. Для чего используются символы ‘ % ’ ? *Остаток от деления*
12. Что обозначает символ ‘ / ’ для целых чисел ? *Целое от Деления*
13. Где указывается тип переменной при ее объявлении? *Перед именем переменной*
14. Чему равно выражение 1 / 2 ? *0*
15. Чему равно выражение 1 % 2 ? *1*
16. Что вычисляет функция x = Math.Abs(A); ? *|А|*
17. Что вычисляет функция x = Math.Round(A); ? *Округляет А*
18. Укажите правильную запись функции возведения X в степень Y в языке C#?

*Math.Pow(x,y)*

1. Что вычисляет функция x = Math.Sqrt(A); ? *Корень из А*
2. Что вычисляет функция Math.Pow(x,y); ? *Возведение в степень*
3. Что вычисляет функция A++; ? *А увеличивается на 1*
4. Чему равно значение переменной x: x = A++; ? *А*
5. Чему равно значение переменной x: x = ++A; ? *А+1*
6. Чему равно значение переменной x: x = – – A; ? *А-1*
7. Вычислить значение x, если int x=8; x += 5; *13*
8. Что вычисляет функция x = Math.Log(A); ? *Логарифм*
9. Какие из следующих последовательностей символов являются операторами присваивания? *Присвоивание происходит при знаке =*
10. Определите назначение символов ‘\t’ в стоке программы:

Console.Write(" {0} \t", a.ToString()); *горизонтальный табулятор*

1. Какое из перечисленных пространств имен классов включает функцию Sqrt?

*Math*.

1. Какое из перечисленных пространств имен включает класс Random? *System*
2. Определите назначение символов ‘\a’ в стоке программы:

Console.Write(" {0} \a", a.ToString()); *Звуковой сигнал*

1. Что означает символ ‘F’ в формате представления чисел Console.WriteLine("x = {0:F} \a", x); языка С# ? *Фиксированная точка*
2. Что означает символ ‘X’ в формате представления чисел Console.WriteLine("x = {0:X} \a", x); языка С# ? *Число в 16ти формате*
3. Что означает символ ‘c’ в формате представления чисел Console.WriteLine("x = {0:c} \a", x); языка С# ? *Денежная единица*
4. Что означает символ ‘E’ в формате представления чисел Console.WriteLine("x = {0:E} \a", x); языка С# ? *Экспонендная форма*
5. Определите назначение символов ‘\n’ в стоке программы:

Console.WriteLine("x = {0} \n", x); *Перевод строки или вертикальный табулятор*

1. Можно ли в языке С# внутри оператора for изменять значение

управляющей переменной? *ДА*

1. Какое значение в языке С# может иметь шаг управляющей переменной в цикле for?

*Любое*

1. Укажите правильную запись цикла for в языке С#? *for(int i=0;i<A;i++){ }*
2. Укажите неправильную запись цикла for в языке С#? *For(i=0;i<A;i++){ }*
3. Какое значение имеет управляющая переменная I после окончания цикла for
   * 1. for (I = 10; I > 1; I--) k = k + 1;
     2. Console.WriteLine("I = {0} ", I);

*1*

1. Как в языке С# выделяются операторы внутри цикла do ... while? *{…}*
2. Когда целесообразно использовать в программе цикл while?

*Когда целесообразней сначала проверить, а потом решить*

1. К какому типу относится цикл while? *Предусловие*
2. С помощью какого оператора можно «досрочно» заканчивать работу цикла?

*Break*

1. С помощью какого оператора можно «пропустить» часть тела цикла?

*Continue*

1. При каком значении "выражения" прекращается выполнение цикла while ("выражение") { . . . } ? *False*
2. При каком значении "выражения" прекращается выполнение цикла

do { . . . } while ("выражение")? *False*

1. Можно ли в «теле» цикла оператора while (I < 5) изменять значение

переменной I? *YES*

1. 100 . . . Укажите правильную запись оператора цикла while? *While(bool){…}*
2. Как должна заканчиваться в языке С# функция если задан ее тип?

*Функция должна возвращать значение return*

1. Где в языке С# задается тип функции ? *Public static int name(){}*
2. Какие переменные в языке С# называются глобальными переменными класса?

*Обьявленный вне метода или функции*

1. Какие переменные в языке С# называются локальными переменными класса?

*Обьявленный в методе или функции*

1. Какие переменные называются формальными параметрами функции?

*Параметры метода (формальные параметры) предназначены для обмена данными между методом и программой*.

1. Какие формальные параметры функции С# называются параметры-ссылки?

*Помеченные служебным словом ref*

1. Какие формальные параметры функции С# называются параметры-значения (входными параметрами)?

*Помеченные служебным словом кроме ref ,оut, params*

1. Какие формальные параметры функции С# называются параметры-массивы?

*Помеченные служебным словом params*

**Задание 2: Решение задач.**

Проверяемые результаты обучения: У1, У2, У3, У4, У5, У6, У7

Обучающимся предлагается решить задачу из предлагаемого перечня:

1. Составить алгоритм вывода таблицы стоимости поездки на такси в рублях, в зависимости от расстояния (1 км стоит 1,5$, доллар равен курсу по ЦБ).
2. Составить алгоритм выбора чисел, меньше заданного числа P, в последовательности квадратов натуральных чисел (1, 4, 9, 25 и т.д.)
3. Составить алгоритм вывода стоимости товаров в чеке до первой суммы, превышающей 1000 руб. Входные данные: S (стоимость). Выходные данные: S.
4. Составить алгоритм поиска всех файлов с именем «к\*т» в каталоге и во всех подкаталогах этого каталога.
5. Составить алгоритм вычисления Z = min (A – B, max (C3, A + B + С)). A, B, C – действительные числа.
6. Составить алгоритм заполнения таблицы данными о зарплате каждого сотрудника за первый квартал 2010 года.

|  |  |  |  |
| --- | --- | --- | --- |
| **ФИО** | **январь** | **февраль** | **март** |
| Иванов | 10000 | 12000 | 13000 |
| Петров | 11000 | 10000 | 12000 |
| Сидоров | 20000 | 21000 | 22000 |
| Васин | 30000 | 31000 | 33000 |

1. Составить алгоритм вычисления общей суммы выплаченной премии всем сотрудникам фирмы. Входные данные: N (количество сотрудников фирмы); аi (размер премии каждого сотрудника); i (параметр). Выходные данные: S (общая сумма выплат).
2. Составить алгоритм вычисления n!. (n!=1\*2\*3\*4\*…\*n) Входные данные: n (натуральное число); i (параметр). Выходные данные: P=n!
3. Составить алгоритм подсчета количества сотрудников фирмы, зарплата которых превышает 20 тыс. руб. (Составить алгоритм подсчета количества чисел больших 20 среди n заданных). Входные данные: n (число сотрудников); S (размер зарплаты); i (параметр). Выходные данные: счетчик К (количество сотрудников).
4. Составить алгоритм табулирования функции y= x2 + 5 на отрезке [a;b]. Значения a, b могут изменяться. Входные данные: х (переменная – параметр цикла); а (начальное значение параметра); b (конечное значение параметра). Выходные данные: y.
5. Составить алгоритм вывода таблицы перевода расстояния в дюймах в сантиметры для значений 10, 11,.... 22 дюйма (1 дюйм = 25,4 мм).
6. Составить алгоритм вывода таблицы перевода перевода 1, 2,... 20 долларов США в рубли по текущему курсу (значение курса вводится произвольно).
7. Составить алгоритм вывода следующих чисел: 1.1, 2.1, …, 21.1.
8. Составить алгоритм вывода стоимости 2, 3, …, 10 кг конфет (цена 1 кг конфет вводится произвольно).
9. Составить алгоритм табулирования функции y =√ x на отрезке [a;b]. Значения a, b могут изменяться.
10. Составить алгоритм табулирования функции y = 1/2x на отрезке [a;b]. Значения a, b могут изменяться.
11. Составить алгоритм табулирования функции y =(x −1)2 на отрезке [a;b]. Значения a, b могут изменяться.
12. Даны два целых числа A и B (A < B). Составить алгоритм вывода всех целых чисел, расположенных между данными числами (не включая сами эти числа), в порядке их возрастания.
13. Даны два целых числа A и B (A < B). Составить алгоритм вывода всех целых чисел, расположенных между данными числами (не включая сами эти числа), в порядке их убывания.
14. Дано целое число N (> 1). Составить алгоритм вывода наименьшего целого K, при котором выполняется неравенство 3K > N и самого значения 3K.
15. Дано целое число N (> 1). Составить алгоритм вывода наибольшего целого K, при котором выполняется неравенство 3K < N.
16. Дано натуральное число N. Составить алгоритм получения всех натуральных чисел, меньше N.
17. Дано число n. Составить алгоритм поиска первого натурального числа, квадрат которого больше n. 54
18. Составить алгоритм вывода минимального числа большее 200, которое нацело делится на 17.
19. Составить алгоритм поиска максимального из натуральных чисел, не превышающих 600, которое нацело делится на 28.
20. Начав тренировки, лыжник в первый день пробежал 10 км. Каждый следующий день он увеличивал пробег на 10 % от пробега предыдущего дня. Составить алгоритм определения, в какой день он пробежит больше 20 км.
21. Составить алгоритм получения двухзначного числа путем перестановки его цифр.
22. Составить алгоритм расчета премии: если оклад сотрудника > 5000 процент премии составляет 10 %; если оклад < 5000, то процент премии равен 12 %.
23. Составить алгоритм нахождения числа десятков и единиц в двухзначном числе.
24. Составить алгоритм вывода названия дня недели по его порядковому номеру (1 – понедельник, 2 – вторник, 3 – среда, 4 – четверг, 5 – пятница, 6 – суббота, 7 – воскресенье).
25. Известны данные о мощности (в единицах силы) двигателей 30 моделей легковых автомобилей. Составить алгоритм определения среди них количества моделей, мощность двигателя которых превышает 200 единиц силы.
26. Группа студентов из 20 человек в сессию сдавала три экзамена. Составить алгоритм заполнения экзаменационной ведомости.
27. Составить алгоритм вычисления следующего выражения .
28. Составить алгоритм вычисления следующего выражения .
29. Составить алгоритм вычисления суммы 

**3. 2 Структура контрольного задания**

**Дифференцированный зачет (5 семестр)**

**Текст типового задания**

**Задание 1:**

Проверяемые результаты обучения: З1,З2,З3,З4

Обучающимся предлагается ответить на два теоретических вопроса из перечня:

1. Как называются формальные параметры функции С#, перед которыми устанавливается служебное слово out?

*– выходные-параметры (помечаются служебным словом* ***out****);*

1. Как называются формальные параметры функции С#, перед которыми устанавливается служебное слово ref?

*параметры-ссылки (помечаются служебным словом* ***ref****);*

1. Что означает служебное слово void перед именем функции? ***void*** *– без типа*
2. Как необходимо использовать функцию в программе, если тип возвращаемого значения функции объявлен void?

*Если перед методом указан тип* ***void****, то метод не должен возвращать результат своей работы с помощью оператора* ***return***

1. Как необходимо использовать функцию в программе, если тип возвращаемого значения функции объявлен double? *С элементом присваивания*
2. Укажите правильную запись заголовка функции

*[ спецификаторы ] тип метода имя метода ( [ параметры ] )*

*{тело метода}*

1. Можно ли внутри функции объявлять другую функцию? *нельзя*
2. Рекурсивной функцией называется функция … *Рекурсия - вызов самой себя*
3. Если некоторая функция в процессе реализации вызывает сама себя, то такая функция называется … *Рекурсивной функцией*
4. Как называется процесс определения нескольких методов с одинаковыми именами? *Перегрузка метода*
5. Чем отличаются «перегруженные» методы?

*Методы с одинаковыми именами, Разные типы формальных параметров*

1. Могут ли «перегруженные» методов имеет формальные параметры разного типа?

*ДА!*

1. Зачем перед именем некоторых методов используется модификатор static?

*К ним можно обращаться “на уровне класса»*

1. Зачем перед именем некоторых методов используется модификатор public?

*Доступна в любом месте программы*

1. Зачем перед именем некоторых методов используется модификатор private?

*Любые методы класса, объявленные со спецификатором* ***private****, доступны только в методах данного класса.*

1. Какой тип параметров может использоваться при перегрузке методов?

*Перегрузка функций часто используется для написания методов, «понимающих» данные различного типа*

1. Какое структурное средство С# позволяет выполнять «декомпозицию» программы?

*декомпозиция – разбиение задачи на отдельные самостоятельные фрагменты.*

1. В основе структурного программирования лежит использование . . .

*Использование фрагментов*

1. Массив это структура данных, содержащая переменные . . . *Одного типа*
2. Можно ли хранить в массиве переменные типа struct? *Можно*
3. Можно ли хранить в массиве переменные, определяемые пользователем?

*Можно (однотипные)*

1. Укажите правильный вариант объявления одномерного массива для 10 переменных целого типа: *Int [] mas=new int[10];*
2. Сколько чисел можно записать в одномерный массив a[15];? *15*
3. Для чего предназначен оператор switch?

*Непосредственно выбор того или иного метода реализации пункта меню осуществляется с помощью оператора* ***switch*** *– переключателя программы*.

1. Как называется оператор, предназначенный для выбора одного из

нескольких вариантов продолжения программы? *switch*

1. Для чего используется служебное слово case в операторе switch?

*возможные варианты продолжения программы с помощью оператора выбора* ***case***

1. Что находится за служебным словом case в операторе switch ?

*Действие или тело! (но если в этой базе встретится в ответах Break подумайте дважды!!!)*

1. Что допускается использовать в <выражение> оператора switch<выражение>?

*Выражение может быть любого типа, для которого существует неявное преобразование к целочисленным типам или типа* ***char*** *(или строкового типа).*

1. Какой тип переменных нельзя использовать в качестве <выражения> в операторе switch<выражения>? *all*
2. Какой тип должка иметь константа выбора в операторе switch<выражение>?

*Выражение может быть любого типа, для которого существует неявное преобразование к целочисленным типам или типа* ***char*** *(или строкового типа).*

1. Что произойдет, если <выражению> в операторе switch<выражение> нет соответствующей константы выбора?

*Если среди констант выбора операторов выбора нет метки совпадающей со значением выражения оператора* ***switch,*** *то программа переходит на выполнение операторов, стоящих за служебным словом* ***default*** *(по умолчанию) или на выполнение оператора, стоящего в программе за оператором* ***switch****.*

1. Каким оператором обычно заканчивается действие метки case в операторе switch?

*«Список операторов» должен заканчиваться оператором перехода, например, оператором* ***break, goto*** *или* ***return.***

1. Почему действие метки case должно заканчиваться оператором break в операторе switch?

*Обычно последним в этом списке операторов находится оператор* ***break****, который завершает работу оператора* ***switch****.*

1. Укажите строку с правильной записью в операторе switch?

*switch ( выражение)*

* 1. *{*
     1. *case константа выбора\_1 : [список операторов; ]*
     2. *case константа выбора\_2 : [список операторов; ]*
        1. *. . .*
     3. *[ default : список операторов; ] }*

1. Что в языке С# означает запись **Unicode** ?

*Двухбайтная система представления символов получила название кодировка* ***Unicode****.*

1. Что в языке С# означает **Unicode**-последовательность?

***Unicode****-последовательностью задается представление символа с помощью его кода.*

1. Для чего в языке C# используются **escape**-последовательности?

*Замена служебных символов для вывода на экран*

1. Какая **escape**-последовательностей осуществляет перевод строки? *\n*
2. Какая **escape**-последовательностей выдает звуковой сигнал (предупреждение)? *\a*
3. Как в программе на языке С# обозначается строковая константа?

*К таким неизменяемым классам относится и класс* ***string****. Ни один из методов этого класса не меняет значения существующих объектов.*

*Неизменность объекта для строковых переменных распространяется и на отдельные его символы – разрешено только чтение отдельных символов, но не их замена.*

1. Для чего используется динамический метод Split при работе со

строковыми переменными?

*Расщеплет текущую строку на лексемы в соответствии с ограничителями delimiters*

1. Что в языке С# делает свойство str.Length, где str - строковая переменная?

*Свойство* ***Length*** *очень часто используется в программах, так как оно позволяет определять «длину» строковой переменной* ***s****, точнее количество символов в строковой переменной .*

1. Что в языке С# делает метод string.Compare(str1, str2), где str1,str2 - строковые переменные?

*Сравнивает две строки и возвращает отрицательное число, если S1 < S2, ноль, если S1 == S2, и положительное число, если S1 > S2. Перегруженные реализации метода позволяют не учитывать разницу в регистре символов, сравнивать подстроки, учитывать особенности национального написания дат, денежных единиц и т. д.*

1. Что в языке С# делает функция str1 = str2.Remove(n1, n2); , где str1, str2 – строковые переменные, а n1, n2 – переменные целого типа?

*Удаляет из текущей строки str2 не более n2 символов, начиная с символа n1*

1. Что в языке С# делает метод str1 = string.Concat(str2, str3); , где str1, str2, str3 – строковые переменные?

*Объединяет строки*

1. Укажите правильный вариант объявления и инициализации двумерного массива

*int[,] a = new int[3,4];*

1. Сколько чисел можно записать в матрицу a, если int[,] a = new int[3,4];?

*12*

1. Что делает следующей фрагмент программы?
2. B = A[0,0];
3. for (j = 0; j <= 10; j++)
4. for (I = 0; I <= 10; I++)
5. if (А[I, j] >b) b = А[I, j];

*поиск максимального элемента*

1. Что делает следующей фрагмент программы?

for (j = 0; j <= 5; j++)

for (I = 0; I <= 10; I++)

if (А[I, j \* 2+1] < 0)

А[I, j \* 2+1] = А[I, j \* 2+1] \* (-1);

*Умножает отрицательные элементы на (-1) в нечетных столбцах*

1. Что делает следующей фрагмент программы

for (I = 0; I <= 10; I++)

for (j = 0; j < 10; j++)

for (k = j +1; k <= 10; k++)

if (a[I,j] > a[I,k])

{ b = a[I,j]; a[I,j] = a[I,k]; a[I,k] = b;} ?

*Сортирует строки в порядке возрастания*

1. Что делает следующей фрагмент программы?

for (I = 0; I<= 10; I++)

for (j = 0; j<= 10; j++)

if (I ==j) a[I, j] = a[I, j] \* (-1); ?

*Главную диагональ умножает на (-1)*

1. Что делает следующей фрагмент программы?

for (I = 0; I <= 10; I++)

if (А[I,I] > 0) А[I,I] = 0;

*Главная диагональ если >0, то приравнять к 0*

1. Что делает следующей фрагмент программы?

for (j = 0; j <= 5; j++)

for (I = 0; I <= 10; I++)

if (А[I,j \* 2] < 0)

А[I,j \* 2] = А[I,j \* 2] \* (-1);

*Если четный столбец <0, то умножить его на (-1)*

1. Как можно уменьшить среднее время поиска в алгоритме блочного поиска?

*Уменьшить размер блока*

1. Какой алгоритм поиска имеет наибольшее среднее время поиска?

*Последовательного поиска*

1. Какой алгоритм поиска имеет наименьшее среднее время поиска?

*Двоичного поиска*

1. Что называется «ключём» поискового массива?

*Значение индекса*

1. Понятие поискового массива?

*Данные, организованные в виде массива*

1. Какой алгоритм поиска чаше всего применяется в базах данных?

*Блочный*

1. Назначение Хеш-функции?

*Хеш-функция преобразует ключ искомого элемента в числовое значение (индекс) в диапазоне от 0 до n-1.*

1. Понятие коллизии при хешированном поиске?

*когда разные ключи получат одинаковые индексы*

1. Какой вариант устранения коллизий чаше других применяется при хешировании?

*использование списочных структур для каждой ячейки хеш-таблицы*

1. Что является основным признаком двунаправленного списка?

*имеются ссылки, как на след.так и на пред. Узел*

1. Что означает циклический список с выделенным заголовком?

*заголовок уст.на элемент не содержит смысловой информации*

1. Какое свойство в списке ArrayList контролирует количество реальных элементов списка?

*Count*

1. Что определяет свойство Count в списке ArrayList?

*св-во предназначено только для чтения, хранит длину*

1. Какой алгоритм имеет большее время поиска, чем алгоритм блочного поиска? *Последовательный*
2. Какой алгоритм имеет меньшее время поиска, чем алгоритм поиска хешированием? *двоичный*
3. Как называется простой список, в котором все включения и исключения элементов выполняются в одном конце списка? *стек*
4. Как называется начало стека? *вершина*
5. Какие операции (из перечисленных ) возможны при работе со стеком? *добавление, удаление, просмотр содержимого*
6. Как называется простой список, в котором все включения и исключения элементов выполняются в любом конце списка? *дек*
7. Какой класс в языке C# специально разработан для работы со стеком? *Stack*
8. Что означаетзаписьPublicStack(intcapacity); ?

*создает пустой стек на capacity элементов*

1. Что означает запись PublicStack(); ?

*создает пустой стек на 10 элементов*

1. Что означаетзаписьPublicStack(ICollectionn); ?

*создает стек на n-элементов*

1. Что означаетзаписьboolstack.Contains(objectv) ?

*возвращает truе, если объект содержится в стеке*

1. Что означает записьvoidstack.Clear() ? *очищает стек*
2. Что означает запись objectstack.Peek() *?возвращает вершины стека, но не удаляет*
3. Что означает запись objectstack.Pop() ? *возвращает вершины стека и удаляет*
4. Что означает запись voidstack.Push(object v)? *Добавляет элементы V в вершину стека*

**Задание 2: Решение задач.**

Проверяемые результаты обучения: У1, У2, У3, У4, У5, У6, У7.

Обучающимся предлагается решить задачу из предлагаемого перечня:

1. С клавиатуры вводятся два числа – количество забитых голов хозяевами и гостями в футбольном матче. Вывести на экран результат игры – победили хозяева/гости/ничья.
2. Напишите программу, которая будет проверять число на кратность 3-м и 7-ми (это числа 21, 42, 63…). Вывести на экран соответствующее сообщение.
3. Вывести на экран 20 элементов последовательности 1, 4, 7, 10, 13…
4. Напишите программу, которая будет «спрашивать» правильный пароль, до тех пор, пока он не будет введен. Правильный пароль пусть будет «root».
5. Дано два массива одинаковой длины (по 10 элементов). Создайте третий массив, который будет отображать сумму первых двух массивов. Первый элемент третьего массива равен сумме первых элементов двух первых массивов и так далее.
6. Используя цикл foreach, выведите на экран все элементы массива целых чисел, которые больше 20 и меньше 50.
7. Напишите функцию, которая будет менять в массиве целых чисел все элементы, которые равны указанному значению (аргумент) на противоположное значение по знаку. Например, все элементы массива которые равны 5, будут меняться на -5.
8. Напишите функцию, которая будет находить минимальное число из трех.
9. Напишите функцию, которая будет возвращать указанный элемент ряда Фибоначчи. Ряд Фибоначчи – это ряд, в котором каждый следующий элемент равен сумме двух предыдущих. 1 1 2 3 5 8 13 21… Функция принимает порядковый номер элемента, и возвращает соответствующий элемент.
10. Дан текст – «Сегодня мы с вами рассмотрели, как работать со строками в Си-шарп. Были описаны основные операторы и методы, которые используются для работы со строками». Обрежьте этот текст так, чтобы осталась только часть «Были описаны основные операторы и методы».
11. Дана строка, которая содержит имена пользователей, разделенные запятой – "Login1,LOgin2,login3,loGin4". Необходимо разбить эту строку на массив строк (чтобы отдельно были логины), и перевести их все в нижний регистр.
12. Есть массив целых чисел размером 10. С клавиатуры вводится два числа - порядковые номера элементов массива, которые необходимо суммировать. Например, если ввели 3 и 5 - суммируются 3-й и 5-й элементы. Нужно предусмотреть случаи, когда были введены не числа, и когда одно из чисел, или оба больше размера массива.
13. Создайте файл numbers.txt и запишите в него натуральные числа от 1 до 500 через запятую.
14. Возьмите любой текстовый файл, и найдите в нем размер самой длинной строки.
15. Дан массив строк: "red", "green", "black", "white", "blue". Запишите в файл элементы массива. Создайте какой-нибудь класс Автомобиль, объявите в нем поля с различными режимами доступа (private, protected, internal, public). Создайте объект класса, и попытайтесь записать/получить данные с различных полей построчно (каждый элемент в новой строке).
16. Создайте класс Телевизор. В нем есть поле текущий канал. Предусмотрите в нем возможность переключения каналов: следующий канал, предыдущий канал, переход к каналу по номеру. Учтите, что канал не может иметь отрицательный номер.
17. Создайте класс Студент, определите в нем поля: имя, курс, есть ли у него стипендия. Создайте в классе несколько конструкторов, для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.
18. Создайте класс Телевизор, объявите в нем поле громкость звука, для доступа к этому полю реализуйте свойство. Громкость может быть в диапазоне от 0 до 100.
19. Создайте базовый класс Геометрическая фигура, предусмотрите в нем общие поля/свойства, например координаты центра фигуры, с помощью конструктора должна быть возможность задать центр. На базе этого класса создайте два новых – Треугольник и Окружность. В этих классах должны быть свои особые поля, например радиус для окружности. В оба класса добавьте метод Нарисовать, в котором могла бы быть специфическая логика рисования фигуры. Создайте объекты треугольник и окружность.
20. Дан двухмерный массив размерностью MxN. Необходимо заполнить его случайными числами в диапазоне от -500 до 500 и отсортировать столбцы матрицы по сумме элементов столбца.
21. Поставить задачу и разработать программу с использованием модификаторов ref и out. Для описания задачи в программе использовать построчные комментарии.
22. Разработать иерархию классов для рисования простых геометрических фигур (звездочками). Необходимо обеспечить возможность отрисовки прямоугольников, треугольников, ромбов (все диагональные линии идут под углом 45 градусов). Оценивается качество и реализация модели.
23. Напишите программу, в которой точка движется справа налево, а затем обратно. При движении цвет точки меняется случайным образом.
24. Написать программу, которая установит, можно ли построить треугольник из отрезков, имеющих длины a, b, c. Если можно, то какой получится треугольник: остроугольный, тупоугольный или прямоугольный. Начальные данные вводятся с клавиатуры, либо передаются в программу как аргументы командной строки.
25. Создать проект, в котором реализуются следующие классы: "Сотрудник", "Фирма". Соответственно, "Фирма" состоит из списка сотрудников, который может изменяться (количество и внутреннее содержимое). Необходимо реализовать следующие возможности: распечатки списка сотрудников при помощи оператора foreach, сравнение сотрудников между собой по должности (продумать возможность хранения табели о рангах), перегрузка операторов + и - для добавления и удаления сотрудников в фирму, соответственно.
26. Разместите на форме четыре кнопки (Button). Сделайте на кнопках следующие надписи: красный, зеленый, синий, желтый. Создайте четыре обработчика события нажатия на данные кнопки, которые буду менять цвет формы в соответствии с текстом на кнопках.
27. Разместите на форме две кнопки (Button) и одну метку (Label). Сделайте на кнопках следующие надписи: привет, до свидания. Создайте обработчики события нажатия на данные кнопки, которые буду менять текст метки, на слова: привет, до свидания. Создайте обработчик события создания формы (Load), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы».
28. Разместите на форме две кнопки и одну метку. Сделайте на кнопках следующие надписи: скрыть, показать. Создайте обработчики события нажатия на данные кнопки, которые буду срывать или показывать метку. Создайте обработчик события создания формы, который будет устанавливать цвет формы и менять текст метки на строку «Начало работы».
29. Разместите на форме три кнопки и одно поле ввода. Сделайте на кнопках следующие надписи: скрыть, показать, очистить. Создайте обработчики события нажатия на данные кнопки, которые буду скрывать или показывать поле ввода. При нажатии на кнопку «очистить» текст из поля ввода должен быть удален.
30. Разработайте игру, которая заключается в следующем. На форме размещены пять кнопок. При нажатии на кнопку какие-то кнопки становятся видимыми, а какие-то невидимыми. Цель игры - скрыть все кнопки.
31. Разработайте игру, которая заключается в следующем. На форме размещены четыре кнопки и четыре метки. При нажатии на кнопку часть надписей становится невидимыми, а часть наоборот становятся видимыми. Цель игры - скрыть все надписи.
32. Разместите на форме ряд кнопок. Создайте обработчики события нажатия на данные кнопки, которые будут делать неактивными текущую кнопку. Создайте обработчик события изменение размера формы (Resize), который будет устанавливать все кнопки в активный режим.
33. Разместите на форме ряд кнопок. Создайте обработчики события нажатия на данные кнопки, которые будут делать неактивными следующую кнопку. Создайте обработчик события нажатия кнопки мыши на форме, который будет устанавливать все кнопки в активный режим.
34. Разместите на форме три кнопки и одно поле ввода. Сделайте на кнопках следующие надписи: \*\*\*\*\*, +++++, 00000. Создайте обработчики события нажатия на данные кнопки, которые будут выводить текст, написанный на кнопках, в поле ввода. Создайте обработчик события создания формы, который будет устанавливать цвет формы и менять текст в поле ввода на строку «Готов к работе».
35. В массиве из 10 целых чисел найти наименьший элемент и поменять его местами с последним элементом.
36. В массиве R, содержащем 25 элементов, заменить значения отрицательных элементов квадратами значений, значения положительных увеличить на 7, а нулевые значения оставить без изменения. Вывести массив R.
37. Дана матрица A(3,4). Найти наименьший элемент в каждой строке матрицы. Вывести исходную матрицу и результаты вычислений.
38. Дана матрица A(3,3). Вычислить сумму второй строки и произведение первого столбца. Вывести исходную матрицу и результаты вычислений.
39. Разработать класс «Записная книжка». В программе должны быть реализованы следующие функции: добавление записи в файл, удаление записи, поиск по признаку (фамилия/имя/город/пол или др.), сортировка по признаку, доступ к записи по номеру вывод количества записей.
40. Разработать класс Matrix (матрица), содержащий следующие методы: конструктор; вывод матрицы; транспонирование матрицы; умножение матрицы на заданное число; сумма двух матриц; разность двух матриц; произведение двух матриц (с проверкой на корректность).

**4. Шкала оценки образовательных достижений**

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

**5. Время на подготовку и выполнение:**

подготовка 10 мин.;

выполнение 1 час 30 мин.;

оформление и сдача 20 мин.;

всего 2 часа 0 мин.

1. **Перечень материалов, оборудования и информационных источников, используемых для аттестации**

Лаборатория вычислительной техники, посадочные места по количеству обучающихся.

1. **Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы**

**Основные источники (печатные издания):**

1. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебник для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2025. — 322 с.

2. Федорова, Г.Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие / Г. Н. Федорова. – М.: КУРС: ИНФРА-М, 2025. – 336 с.

3. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 196 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.МайоДж. С# Builder. Быстрый старт. — М.: Бином, 2005. - 384 с.

5.Саша Годштейн, Дима Зурбалев, 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: 978-5-496-00867-9 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.Оптимизация приложений на платформе .NET

Саша Годштейн, Дима Зурбалев, 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>