Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Оренбургский государственный университет»**

Кафедра прикладной информатики в экономике и управлении

**Методические указания**

*по дисциплине*

*«Б1.Д.Б.11 Информатика и программирование»*

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

*09.03.03 Прикладная информатика*

(код и наименование направления подготовки)

*Прикладная информатика в экономике*

(наименование направленности (профиля) образовательной программы)

Квалификация

*Бакалавр*

Форма обучения

*Очная*

Год набора 2021

Составитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вдович С.А., Панова Н.Ф.

Методические указания рассмотрены и одобрены на заседании кафедры прикладной информатики в экономике и управлении

## Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Жук М.А.

Методические указания является приложением к рабочей программе по дисциплине «Информатика и программирование», зарегистрированной в ЦИТ под учетным номером \_\_\_\_

**Содержание**

|  |  |
| --- | --- |
| 1 Методические указания по лекционным занятиям ………………............................ | 4 |
| 2 Методические указания по лабораторным работам …………………………………. | 5 |
| 3 Методические указания по выполнению ИТЗ………………………………………… | 6 |
| 4 Методические указания по промежуточной аттестации по дисциплине……………. | 7 |

## 1 Методические указания по лекционным занятиям

Лекции являются одним из основных методов обучения по дисциплине «Информатика и программирование» , решающим следующие задачи:

* изложить важнейший материал программы курса;
* познакомить с основными понятиями информатики, способами кодирования информации и представления данных в памяти ЭВМ;
* развивать у обучающихся потребность к самостоятельной работе над учебниками и научной литературой.

Изучение дисциплины следует начинать с проработки рабочей программы, особое внимание необходимо уделить целям и задачам, структуре и содержанию курса.

При конспектировании лекций обучающимся, необходимо излагать услышанный материал на лекции своими словами. Необходимо выделять важные места в своих записях. Каждый раз, когда что-либо не понятно, необходимо записывать свои вопросы. По возможности можно сравнивать свои конспекты с конспектами двух-трех других обучающихся, при этом дополняя и исправляя свои записи.

Успешное освоение курса предполагает активное, творческое участие обучающегося путем планомерной, повседневной работы.

Лекционный материал необходимо кратко записывать, обращая внимание, на логику изложения материла, аргументацию и приводимые примеры.

Лекционный материал следует просматривать в тот же день, когда читалась лекция, помечая непонятные места. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь за консультацией на ближайшей лекции к преподавателю.

Рекомендуемую дополнительную литературу следует прорабатывать после изучения данной темы по учебнику и материалам лекции.

При подготовке материала необходимо обращать внимание на точность определений, последовательность изучения материала, аргументацию, собственные примеры.

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

## 2 Методические указания по выполнению лабораторных работ

Выполнение лабораторных работ обучающимися является необходим условием успешного освоения дисциплины «Информатика и программирование». Выполнение лабораторных работ способствует укреплению теоретического материала и освоению практических навыков решения экономических и расчетных задач с применением современных средств обработки данных.

Программа курса, согласно рабочей программе, включает несколько лабораторных работ, охватывающих весь учебный курс. Каждой лабораторной работе предшествует теоретический материал и контрольные вопросы по рассматриваемой теме. Лабораторная работа имеет следующую структуру: тема, цели, краткие теоретические сведения, порядок проведения работы, требования к составлению отчета.

Перед началом выполнения лабораторной работы необходимо ознакомиться с целью и содержанием работы, изучить требуемый для выполнения работы теоретический материал.

После выполнения лабораторной работы студент должен представить отчет о проделанной работе. Оценку по практической работе студент получает, если студентом работа выполнена в полном объеме, студент может пояснить выполнение любого этапа работы, отчет выполнен в соответствии с требованиями к выполнению работы, студент отвечает на контрольные вопросы на удовлетворительную оценку и выше.

Зачет по выполнению лабораторных работ студент получает при условии выполнения всех предусмотренных программой лабораторных работ с отчетами по всем работам.

Отчет к лабораторной работе должен содержать:

- Тему работы;

- Задание для выполнения, включая индивидуальное задание;

- Результаты выполненного задания;

- Выводы.

**3 Методические указания по выполнению индивидуального творческого задания**

Самостоятельная учебная деятельность является необходимым условием успешного обучения. Многие профессиональные навыки, способность мыслить и обобщать, делать выводы и строить суждения, выступать и слушать других, – все это развивается в процессе самостоятельной работы обучающихся.

Изучение дисциплины «Информатика и программирование» предполагает не только познавательную деятельность, которую обучающиеся выполняют во время лабораторных занятий, но и самостоятельную работу, осуществляемую вне аудиторных занятий.

Самостоятельная работа в форме ИТЗ является индивидуальной самостоятельно выполненной работой обучающегося. Индивидуальное задание способствует формированию навыков самостоятельного научного творчества, повышению теоретической и профессиональной подготовки, лучшему усвоению учебного материала, является одной из форм самостоятельной работы обучающихся. В первом семестре ИТЗ представляет собой решение задачи в одной из табличных процессоров, например MS Excel с применением функций и макросов. Обучающемуся необходимо:

1 Реализовать решение задачи в пакете MS Excel согласно варианту

2 Оформить отчет по ИТЗ, содержащий постановку задачи, метод решения (с использованием функций, макросов и др.), экранную форму исходных данных и результата.

Варианты заданий представлены в фонде оценочных средств данной дисциплины.

Во втором семестре целью индивидуального творческого задания является приобретение навыков разработки программ на языке C#. При выполнении индивидуального задания студент должен показать умение самостоятельно разрабатывать алгоритмы и программы.

Результат выполнения индивидуального задания оформляется в виде отчета с приложенной программной реализацией и скриншотами. Основой для успешного выполнения индивидуального задания является изучение лекционного материала и регулярное выполнение лабораторных работ.

Индивидуальное задание – самостоятельная творческая завершенная работа студента, выполненная под руководством преподавателя. Основные цели работы: развитие активной творческой личности, способной самостоятельно приобретать новые знания, развитие навыков самостоятельной исследовательской работы у студентов.

Выполнение задания состоит из нескольких этапов:

1. Разработка алгоритма;
2. Написание программы;
3. Отладка и тестирование программы.

Этапы задания выполняются последовательно, после изучения соответствующих тем дисциплины.

Защита работы студентом осуществляется после завершения последнего этапа. Отчет предоставляется в распечатанном виде, разработанное приложение – в электронном виде.

**5 Методические указания по промежуточной аттестации по дисциплине**

Основная цель промежуточной аттестации - завершение обучения по конкретной дисциплине путем проверки уровня приобретенных обучающимися знаний.

При подготовке к промежуточной аттестации обучающемуся необходимо:

- обратиться к пройденному материалу – повторить и закрепить знания, приобретенные в период изучения учебной дисциплины;

- изучить дополнительные источники информации для получения новых знаний, в том числе вновь изданные учебники и пособия.

При этом необходимо учитывать, что при проведении промежуточной аттестации проверяется не только способность обучающегося воспроизвести изученный им материал, но и то, насколько обучающийся понимает данный материал, умеет анализировать его, имеет свое собственное мнение и умеет отстаивать его посредством грамотного обоснования.

Итоговым контролем при изучении дисциплины «Информатика и программирование» является экзамен. Примерный перечень вопросов к экзамену содержится в фонде оценочных средств дисциплины. Указанные вопросы по дисциплине «Информатика и программирование» обновляются на начало учебного года. На экзамене обучающемуся предлагается ответить на два вопроса по изученным разделам дисциплины и решить задачу. Цель экзамена - проверка и оценка уровня полученных обучающимся специальных познаний по учебной дисциплине «Информатика и программирование», а также умения логически мыслить, аргументировать избранную научную позицию, реагировать на дополнительные вопросы, ориентироваться в массиве программного обеспечения. Оценке подлежит также и правильность речи обучающегося. Дополнительной целью итогового контроля в виде экзамена является формирование у обучающегося таких качеств, как организованность, ответственность, трудолюбие, самостоятельность.

При подготовке к экзамену обучающийся должен правильно и рационально распланировать свое время, чтобы успеть качественно и на высоком уровне подготовиться к ответам по всем вопросам. Во время подготовки к экзамену обучающиеся также систематизируют знания, которые они приобрели при изучении разделов курса.

Рекомендуемые учебники и специальная литература при изучении курса «Информатика и программирование», имеются в рекомендованном списке литературы в рабочей программе по данному курсу.

Обучающийся в целях получения качественных и системных знаний должен начинать подготовку к экзамену задолго до его проведения, лучше с самого начала лекционного курса. Для этого, как уже отмечалось, имеются в учебно-методическом пособии примерные вопросы к экзамену.

Самостоятельная работа по подготовке к экзамену во время сессии должна планироваться обучающимися, исходя из общего объема вопросов, вынесенных на экзамен и дней, отведенных на подготовку. При этом необходимо, чтобы последний день или часть его, был выделен для дополнительного повторения всего объема вопросов в целом. Это позволяет обучающемуся самостоятельно перепроверить уровень усвоения материала.

На правах рукописи

Минобрнауки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Оренбургский государственный университет»**

Кафедра прикладной информатики в экономике и управлении

Методические указания по лабораторным работам для обучающихся

по дисциплине

*«Б.1.Б.11 Информатика и программирование»*

Уровень высшего образования

БАКАЛАВРИАТ

*09.03.03 Прикладная информатика*

(код и наименование направления подготовки)

*Прикладная информатика в экономике*

(наименование направленности (профиля) образовательной программы)

Тип образовательной программы

*Программа академического бакалавриата*

Квалификация

*Бакалавр*

Форма обучения

*Очная*

Год набора 2021

Составители Вдович С.А., Панова Н.Ф

Методические указания рассмотрены и одобрены на заседании кафедры прикладной информатики в экономике и управлении

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Жук М.А.

Методические указания является приложением к рабочей программе по дисциплине «Информатика и программирование», зарегистрированной в ЦИТ под учетным номером\_\_\_\_\_\_\_\_\_\_\_

# 

Оглавление

[1. Лабораторная работа №1 Освоение операционной среды Windows 11](#_Toc22734914)

[2. Лабораторная работа №2 Основы работы с текстовым редактором MS Word 11](#_Toc22734915)

[3. Лабораторная работа №3. Основы работы с текстовым редактором MS Word: таблицы 12](#_Toc22734916)

[4. Лабораторная работа №4. Основы работы с текстовым редактором MS Word: графика 14](#_Toc22734917)

[5. Лабораторная работа №5. Электронные таблицы MS Excel 15](#_Toc22734918)

[6. Лабораторная работа №6. Основы работы в среде локальных и глобальных компьютерных сетей 22](#_Toc22734919)

[7. Лабораторная работа №7. Основные сведения о платформе Microsoft .NET. Первая программа на C# 22](#_Toc22734920)

[8. Лабораторная работа №8. Программы линейной структуры. Класс Math 23](#_Toc22734921)

[9. Лабораторная работа №9. Программы ветвящейся структуры. Оператор If 23](#_Toc22734922)

[10. Лабораторная работа №10. Оператор switch 24](#_Toc22734923)

[11. Лабораторная работа №11. Циклы с известным числом повторений. Итерационные циклы. Сложные циклы 25](#_Toc22734924)

[12. Лабораторная работа №12. Обработка одномерных массивов. Обработка матриц 25](#_Toc22734925)

[13. Лабораторная работа №13. Обработка строк 26](#_Toc22734926)

[14. Лабораторная работа №14. Разработка программ с использованием классов 27](#_Toc22734927)

# 1. Лабораторная работа №1 Освоение операционной среды Windows

* 1. **Цель работы:** приобретение навыков работы в ОС Windows: Управление ресурсами системы средствами Windows Explorer. Настройка индивидуальной среды пользователя, комплекс средств Control Panel.
  2. **Задания:**
  3. 1 Регистрация пользователя в системе
  4. 2 Изучение папок профиля пользователя и библиотек
  5. 3 Настройка компонентов рабочего стола (меню кнопки Пуск, значки панели задач, ярлыки рабочего стола, гаджеты)
  6. 4 Настройка и использование проводника, Работа с папками и файлами.
  7. 5 Способы запуска программ и открытия папок и файлов.
  8. **Контрольные вопросы:**

1 Что отображается на экране сразу после загрузки операционной системы Windows?

2 Способы запуска программ

3 Что определяет расширение файла

4 Какие действия приведут к перемещению файла

5 Как перейти к строке меню программы?

6 Что произойдет с папкой, если удалить ее ярлык с рабочего стола?

7 Как выделить в списке подряд идущие файлы?

8 Как перейти из папки на один уровень вверх?

9 Укажите правильные способы удаления файла

10 Как с помощью клавиатуры запустить справку в стандартной программе Windows?

11 Что содержит вкладка «Указатель» справочной системы?

12 Как отобразить панель быстрого запуска?

13 С помощью какой стандартной программы можно произвести проверку поверхности жесткого диска в Windows ?

14 Как отформатировать съемный диск?

# 2. Лабораторная работа №2 Основы работы с текстовым редактором MS Word

* 1. **Цель работы:** Освоить технику ввода и форматирование текста. Создание списков, сносок, ссылок в MS Word.
  2. Теоретическая часть:
  3. ***Выделение произвольного количества символов.***

Нажмите левую кнопку мыши на первом символе фрагмента и, не отпуская кнопку, растяните выделение до конца фрагмента.

***Выделение слова.***

Дважды щелкните левой кнопкой мыши на выделяемом слове.

***Выделение нескольких строк.***

Поместите указатель мыши **СЛЕВА** от строки. Указатель должен превратиться из вертикальной черточки в стрелку. Теперь нажмите левую кнопку мыши, и, не отпуская кнопку, переместите мышь вертикально вниз (или вверх) для выделения нескольких строк.

***Выделение фрагмента при помощи клавиатуры.***

Поставьте курсор (текстовый курсор, а не указатель мыши!) **ПЕРЕД** первым символом фрагмента. Нажмите клавишу **[Shift]** и, не отпуская ее, с помощью курсорных клавиш-стрелок растяните выделение на весь фрагмент.

***Настройка отступов после абзацев.***

В образце приглашения вы можете заметить, что между некоторыми абзацами есть пустое пространство. Эти промежутки можно было сделать при помощи пустых строк – нажимая лишний раз **[Enter]** при вводе текста. Этот способ неудобен тем, что размер промежутка, если потребуется, будет довольно сложно изменить – каждый придется менять по отдельности. Удобнее сделать отступы с помощью специального свойства абзаца – **Интервал после**.

**Задания:**

**1** Выполнить оформление документа Приглашение

**2** Осуществить форматирование реферата, по следующим критериям:

- каждый раздел с нового листа (содержание, введение, заключение, отдельные разделы, список использованных источников);

- шрифт основного текста Times New Roman 14;

- шрифт заголовков разделов Times New Roman 16 полужирный;

- межстрочный интервал одинарный;

- выравнивание по ширине;

- отступ первой строки 1,25, все поля 2 см.

- номера страниц справа внизу листа;

- перечисления оформляются с абзацного отступа через символ "-";

- предусмотреть наличие сносок;

- использовать «Колонтитул» внизу каждой страницы хранить ФИО и название реферата.

- используя средства редактора MS Word автоматизировать создание содержания.

**3 Списки**

Сформируйте список вашей группы и оформите его в виде списка. Для оформления многоуровневого списка используйте один из следующих вариантов:

комбинацию клавиш shift + Alt + стрелки влево или вправо (клавиши управления курсором), предварительно установив курсор в нужную строку;

предварительно выделив нужную строку, вызовите контекстное меню и выберите нужную команду (понизить уровень или повысить уровень).

# 3. Лабораторная работа №3. Основы работы с текстовым редактором MS Word: таблицы

* 1. **Цель работы:** Освоить работу с таблицами, вычисляемыми таблицами, редактор формул.
  2. **Теоретическая часть:**

Таблица может содержать максимум 31 столбец и произвольное число строк, таблицы имеют адреса, образованные именем столбца (А, В, С...) и но-(1, 2, 3...), например, А1, С4 и т.д. Word позволяет выполнять вычисления, записывая в отдельные ячейки таблицы формулы с помощью команды ***Таблица\Формула...*** Формула задается как выражение, в котором использованы:

*абсолютные* ссылки на ячейки таблицы в виде списка (разделяемые знаком «;» А1; В5; Е10 и т.д.) или блока (начало и конец блока ячеек — **А1:А10);**

*ключевые* слова для ссылки на блок ячеек:

LEFT — ячейки, расположенные в строке левее ячейки с формулой; RIGHT — ячейки, расположенные в строке правее ячейки с формулой-ABOVE — ячейки, расположенные в столбце выше ячейки с формулой-BELOW — ячейки, расположенные в столбце ниже ячейки с формулой;

константы — числа, текст в двойных кавычках;

закладки, которым соответствует определенный текст документа, созданный с помощью команды ***Правка\3акладка;***

встроенные функции Word, например, SUM().

знаки операции (+, -, \*, /, %, =, <, >, <=, >=).

**Задания:**

**1** Создать таблицу:

**Сведения об обеспечении образовательного процесса учебной литературой в учебном центре ТОО «МаксСофт»**

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование предмета (курса, учебного плана) по годам обучения | Число  обучающихся,  воспитанников,  изучающих  предмет (курс, дисциплину) | Обеспечение обучающихся литературой | |
| Перечень литературы | Число экземпляров |
| 1 | 2 | 3 | 4 |
| Курсы для пользователей | 1000 | А. Бабий. компьютер куплен. | 1000 |
| Курсы для программистов | 10 | Фирменная документация | 1 |

**2** Создать таблицу, произвести вычисления успеваемости студентов.

Сведения об успеваемости студентов общеэкономического факультета КГУ за 2017/18 учебный год

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № пп | Учебная дисциплина | Группа | Средний  балл | Всего  сдавало | Отл. | Хор. | Удовл. | Неудоап. | Неявки |
|  | Высшая математика |  |  |  |  |  |  |  |  |
| 1 |  | 51 |  | ячейка  Е4 | 12 | 10 | 6 | 3 | 1 |
| 2 |  | 52 |  |  | 7 | 9 | 6 | 3 | 2 |
| 3 |  | 53 |  |  | 9 | 8 | 3 | 5 | 3 |
| 4 |  | 54 |  |  | 8 | 8 | 8 | 3 | 2 |
|  | Итого: |  |  |  |  |  |  |  |  |

Технология выполнения:

Введите формулы для расчета числа студентов каждой группы, сдавших экзамен по дисциплине «Высшая математика». Для этого установите курсор в ячейку  
Е4 и введите формулу: SUM(RIGHT), предварительно убрав имеющуюся в ней  
запись. Ввести формулу надо, используя меню ***Таблица\Формула....***

Произведите те же действия для ячеек Е5 — Е7.

Введите формулу для расчетов среднего балла по дисциплине «Высшая математика» для группы 51. Для этого установите курсор в ячейку D4 и введите формулу: =(F4\*5+С4\*4+Н4\*3+I4\*2)/Е4. Выберите формат числа 0,00.

Введите аналогичные формулы в ячейки D5-D7.

Введите формулу для расчета общего числа студентов, сдавших экзамен  
по каждой дисциплине на отлично, хорошо и т.д. Для этого установите курсор  
в ячейку F8 и введите формулу: SUM(ABOVE), затем аналогично в ячейку G8.

**3** Используя редактор формул наберите фрагмент из учебника математики по указанной в варианте теме, содержащий математические формулы.

1 Теория пределов.

2. Непрерывность функции.

3. Производная.

4. Дифференциал функции

5. Правила дифференцирования.

6. Производные элементарных функций.

7. Неявные функции.

8. Неопределенный интеграл.

9. Элементы таблицы интегрирования элементарных функций.

10. Определенный интеграл.

11. Основная теорема интегрального исчисления.

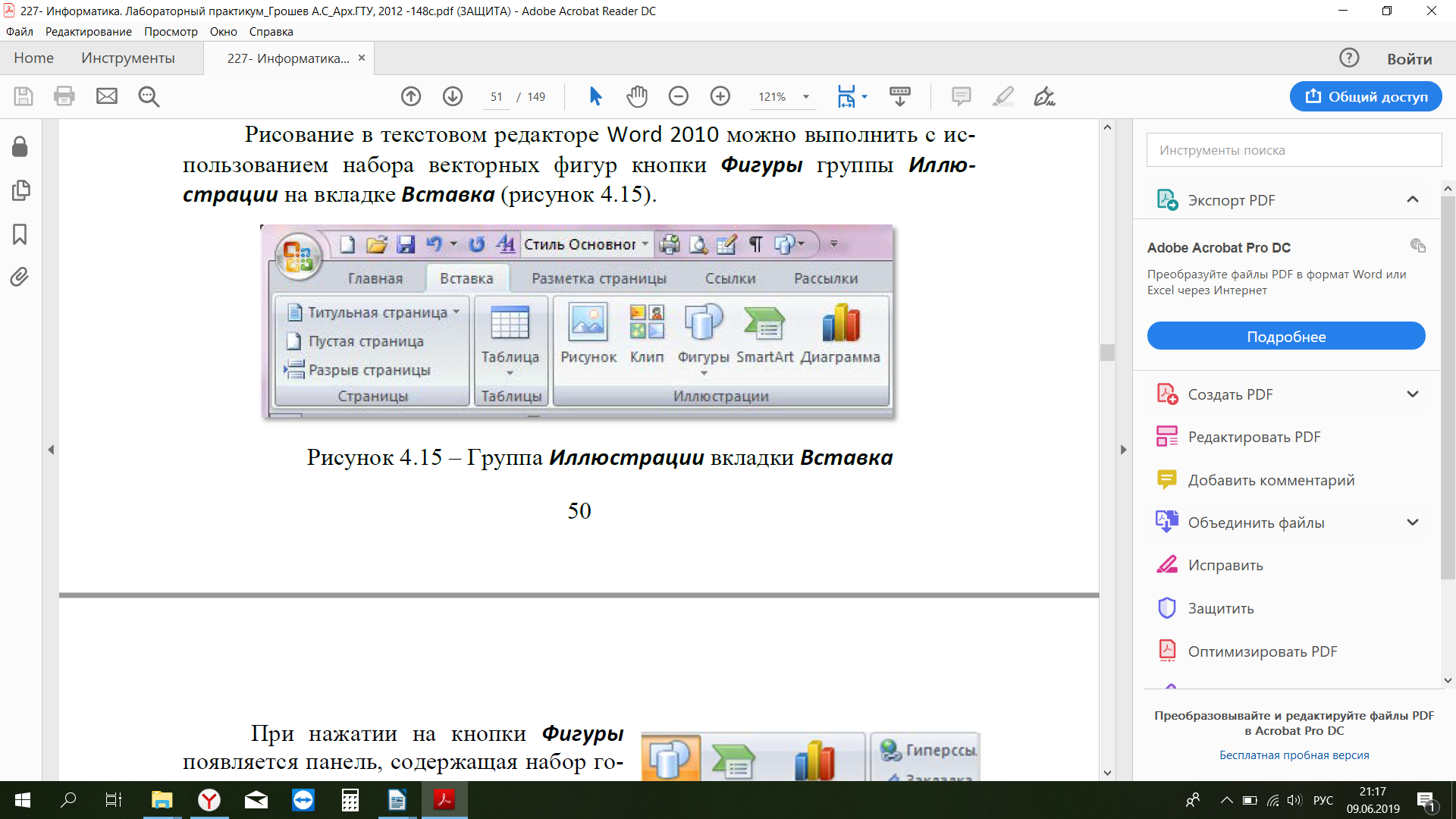
12. Двойной интеграл.

13. Ряды Тейлора.

14. Интерполирование по Лагранжу.

15. Интерполирование по Ньютону.

# 4. Лабораторная работа №4. Основы работы с текстовым редактором MS Word: графика

* 1. **Цель работы:** Создание блок-схем, рисунков, графических изображений.
  2. **Теоретическая часть:** рисование в текстовом редакторе можно выполнить с использованием набора векторных фигур кнопки Фигуры группы Иллюстрации на вкладке Вставка
  3. 
  4. **Задание:**
  5. Оформить блок-схемы согласно выданному варианту.
  6. **Контрольные вопросы:**

1 Текстовый процессор Ms Word: основные приемы работы, краткая характеристика пунктов меню

2 Текстовый процессор Ms Word: работа с текстом (настройка шрифта, параметров страницы, метода выравнивания, параметров абзаца, вставка надстрочных и подстрочных символов, зачеркивание слов)

3 Текстовый процессор Ms Word: создание маркированных и нумерованных списков, сноски, колонтитулы, номера страниц, ссылки, перекрестные ссылки, заголовки.

4 Текстовый процессор MS Word: редактор формул

5 Текстовый процессор MS Word: создание таблиц (форматирование строк и столбцов, заливка ячеек, объединение ячеек, выравнивание ячеек)

6 Текстовый процессор MS Word: вычисляемые таблицы

7 Текстовый процессор MS Word: приемы работы с объектами

8 Текстовый процессор MS Word: приемы работы с рисунками (вставка надписи, автофигуры; форматирование: яркость, контрастность, положение на странице, обрезка и т.д.; работа с композициями автофигур: группировка, настройка порядка, перемещение одновременно нескольких объектов, выравнивание фигур относительно друг друга и т.д.)

9 Текстовый процессор MS Word: предварительный просмотр текста, печать, управление масштабом, режимы просмотра документа, использование линейки, сетки, схема документа, поиск и замена слов, быстрый переход на нужную страницу, раздел документа и т.д.

10 Рецензирование текста: проверка правописания, перевод текста, тезаурус, создание примечаний, сносок, защита документа, расстановка переносов.

11 Кнопка office: отправка текста, свойства, проверка совместимости и т.д.

12 Текстовый процессор MS Word:оглавление и указатели.

# 5. Лабораторная работа №5. Электронные таблицы MS Excel

**Цель работы:**  приобрести навыки использования электронных таблиц для решения задачи профессиональной области.

**Теоретическая часть**

Общие понятия

*Электронная таблица* **–** совокупность строк и столбцов.

*Столбцы* обозначены буквами латинского алфавита, а *строки* - цифрами. Пересечение строки и столбца называется *клеткой* или *ячейкой*.

Каждая клетка имеет свой *адрес*, который состоит из буквы столбца и цифры строки (А5, В7, К4 …).

Задание 1. Элементы окна Excel.

Запустите Excel. На экране появится окно Excel с открывшейся пустой рабочей книгой под названием *Книга 1*.

Окно Excel содержит многие типичные элементы окон Windows. Внимательно рассмотрите окно Excel и выделите элементы интерфейса:

Строка заголовка,

Строка меню

Панели инструментов – по умолчанию окно содержит две панели: стандартную и форматирования. Чтобы отобразить другие панели инструментов или скрыть не используемые в работе, используйте команду *Вид/Панели инструментов*.

Строка формул

Заголовок столбцов – буквы латинского алфавита, расположенные в верхней строке листа и определяющие столбцы рабочей таблицы. Колонки нумеруются в следующем порядке: A – Z, AA – AZ, BA – BZ и т.д.

Заголовок строки – числа, расположенные по левому краю листа и определяющие строки рабочей таблицы.

Окно рабочей книги

Ярлыки листов рабочей таблицы – для перехода от одного листа к другому в пределах рабочей книги.

Строка состояния

**В ячейку можно ввести**

**значение (константу)** одного из четырех типов:

текст,

число,

дата-время,

логическое значение (Истина, Ложь)

**формулу**, результатом которой будет являться значение одного из перечисленных выше типов или ошибочное значение.

Для ввода данных **необходимо:**

выделить нужную ячейку,

ввести значение или формулу,

нажать клавишу Enter.

Редактирование данных в активной ячейке **можно выполнить тремя способами (для завершения редактирования всегда нужно нажать клавишу Enter):**

щелкнув в строке формул, внести в этой строке исправления,

нажать F2 и внести исправления непосредственно в ячейке,

дважды щелкнуть по ячейке и внести исправления непосредственно в ячейке.

Примечание. Для исправления непосредственно в ячейке должен быть установлен флажок *Правка* *прямо в ячейке* (*Сервис/Параметры/*вкладка *Правка*)

Удаление содержимого **ячейки:**

выделить ячейку или группу ячеек,

нажать клавишу Delete или выполнить команду меню *Правка/Очистить/Содержимое*.

Изменение ширины столбца и высоты строки**:**

подведите указатель мыши на разделительную линию между именами столбцов или строк,

указатель примет вид двойной стрелки,

нажмите левую кнопку мыши и растяните (сожмите) столбец или строку.

Вставка строки или столбца**:**

щелкните на ячейке в столбце, слева от которого следует вставить новый столбец, или на ячейке в строке, над которой следует вставить новую строку,

откройте меню Вставка и выберите команду Строки или Столбцы, после этого программа вставит новые строки или столбцы.

Примечание. Чтобы быстро вставить один или несколько столбцов или строк выделите нужное количество столбцов или строк, щелкните на них правой кнопкой мыши и в появившемся контекстном меню выберите пункт *Вставить*.

Свойства (форматирование) ячейки**:**

тип и цвет фона,

тип, размер и цвет шрифта,

тип, цвет и место проведения границы,

тип выравнивания,

тип ориентации текста,

числовой формат,

защита.

Чтобы **установить свойства ячейке** или группе ячеек, необходимо выделить эту ячейку (группу) и выбрать в меню команду *Формат/Ячейки* или нажать клавиши быстрого вызова **Ctrl-1**.

Чтобы **отменить ранее установленные свойства** ячейки необходимо выполнить команду верхнего меню Правка/Очистить/Форматы.

Чтобы **удалить содержимое и отменить ранее установленные свойства**, нужно выполнить команду меню Правка/Очистить/Все

**Относительные и абсолютные ссылки**

**Относительная ссылка** используется в формуле для указания адреса ячейки, вычисляемого в относительной системе координат с началом в текущей ячейке.

Относительные ссылки имеют следующий вид: А1, В1

**Абсолютная ссылка** используется для указания адреса ячейки, вычисляемого в абсолютной системе координат и не зависящего от текущей ячейки.

Абсолютные ссылки имеют вид: $А$5, $F$5, $G$3 и т.п.

Примечание

Для того, чтобы относительную ссылку преобразовать в абсолютную, достаточно после ввода ссылки нажать клавишу **F4** – и знаки доллара появятся автоматически.

|  |  |  |
| --- | --- | --- |
| Название ссылок | Обозначения | Результат при копировании или переносе формул |
| Частичная абсолютная ссылка | $F5 | Не меняет номер столбца |
| F$5 | Не меняет номер строки |
| Полная абсолютная ссылка | $F$5 | Не меняет номер строки, номер столбца |

**Задания:**

**Вычисления, использование абсолютных и относительных ссылок**

**№ 1**

Подготовьте таблицу для начисления пени в соответствии с образцом.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Оплата коммунальных услуг задержана на** | | |  | **дней** |
|  | | | | |
| **Вид оплаты** | **Начисленная сумма** | **Пени** | **Всего к оплате** | |
| Квартплата |  |  |  | |
| Газ |  |  |  | |
| Электричество |  |  |  | |
| Телефон |  |  |  | |
|  | | | | |

**Пени** высчитывается по формуле – 1% от начисленной суммы за каждый задержанный день.

**Всего к оплате** считается как сумма начисления плюс пени.

Добавьте в таблицу строку для подсчета итоговых показателей:

всего начислено, всего пени, всего к оплате.

**№2**

Представьте себя одним из организаторов игры «Угадай мелодию». Вам поручено отслеживать количество очков, набранных каждым игроком, и вычислять суммарный заработок в рублях в соответствии с текущим курсом валюты.

Заготовьте таблицу по образцу.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Курс валюты** | | |  |  |
|  | | | | |
|  | 1 раунд | 2 раунд | 3 раунд | **Суммарный заработок за игру  (в руб.)** |
| 1-й участник |  |  |  |  |
| 2-й участник |  |  |  |  |
| 3-й участник |  |  |  |  |
|  | | | | |

Примечание

Формулу для определения суммарного заработка пишите только для первого игрока, а остальным её распространяете.

**№3**

Составьте таблицу сложения чисел первого десятка.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица сложения | | | | | | | | | | |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |

**№4**

Вычислить сумму n-го члена арифметической прогрессии: an = a1+d(n-1)

Sn = (a1+an)\*n/2

Составить таблицу:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | А | В | С | 0 |
| 1 | Вычисление *п-го* члена и суммы арифметической прогрессии | | | |
| 2 |  | | | |
| 3 | d | n | an | Sn |
| 4 | 0,725 | 1 | -2 | -2 |
| 5 | 0,725 | 2 | -1,275 | -3,275 |
| 6 | 0,725 | 3 | -0,55 | -3,825 |
| 7 | 0,725 | 4 | 0,175 | -3,65 |
| 8 | 0,725 | 5 | 0,9 | -2,75 |
| 9 | 0,725 | 6 | 1,625 | -1,125 |
| 10 | 0,725 | 7 | 2,35 | 2,225 |
| 11 | 0,725 | 8 | 3,075 | 4,3 |
| 12 | 0,725 | 9 | 3,8 | 8,1 |
| 13 | 0,725 | 10 | 4,525 | 12,625 |

**№ 5 «Фильтрация данных»**

**Дана таблица**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ФИО** | **Группа** | **Оценки по дисциплинам** | | | **Ср. бал** | **Факультет** |
| **Информатика** | **Математика** | **Иностранный язык** |
|  |  |  |  |  |  |  |

Заполнить не менее 15 строк. Посчитать средний бал. Осуществить:

1 Сортировку данных по алфавиту по полю ФИО

2 Скопировать таблицу на второй лист, строки, в которых средний балл ниже 3,5 окрасить в красный цвет.

3 Скопировать таблицу на третий лист осуществить сортировку данных по столбцу средний балл от наибольшего к наименьшему значению.

4 Скопировать таблицу на четвертый лист, выделить желтым цветом обучающихся, у которых все оценки 5.

**Вопросы:**

Электронные таблицы MS Excel: основные приемы работы, краткая характеристика пунктов меню

Электронные таблицы MS Excel: основные понятия, краткая характеристика пунктов меню

Электронные таблицы MS Excel: типы ссылок и копирование данных

Электронные таблицы MS Excel: формулы

Электронные таблицы MS Excel: форматирование и оформление данных

Электронные таблицы MS Excel: автозаполнение по образцу

Электронные таблицы MS Excel: функции

Электронные таблицы MS Excel: построение графиков и диаграмм

Электронные таблицы MS Excel: использование надстроек

# 6. Лабораторная работа №6. Основы работы в среде локальных и глобальных компьютерных сетей

* 1. **Цель работы:**  освоить навыки поиска справочной информации с применением информационно-коммуникационных технологий.
  2. **Задание:**
  3. **1** Изучить популярные браузеры, составить их сравнительную характеристику.
  4. **2** Найти информацию для реферата, согласно предложенной теме. Оформить реферат согласно требованиям оформления студенческих работ.

# 7. Лабораторная работа №7. Основные сведения о платформе Microsoft .NET. Первая программа на C#

* 1. Для выполнения лабораторных работ во втором семестре используется Средства для разработки и проектирования Microsoft Visual Studio. Результаты выполнения лабораторных работ представляются в виде отчета, содержащего блок-схему выполнения программы, исходный текст программы и скриншот с результатами.

Платформа .NET Framework — это управляемая среда выполнения для ОС Windows, предоставляющая разнообразные службы запускаемым в ней приложениям. Она состоит из двух основных компонентов: среды CLR — механизма, управляющего выполняющимися приложениями, и библиотеки классов .NET Framework — библиотеки проверенного кода, предназначенного для повторного использования, который разработчики могут вызывать из своих приложений. Вот какие службы предоставляет .NET Framework выполняемым в ней приложениям:

* Управление памятью
* Общая система типов.  В .NET Framework базовые типы определяются системой типов .NET Framework, при этом для всех языков .NET Framework используются одни и те же базовые типы.
* Обширная библиотека классов. Разработчикам не требуется писать код для выполнения стандартных низкоуровневых операций программирования, так как они используют удобную библиотеку типов и членов, входящую в библиотеку классов .NET Framework.
* Платформы и технологии разработки. Платформа .NET Framework включает библиотеки для конкретных областей разработки приложений, например ASP.NET для веб-приложений, ADO.NET для доступа к данным
* Взаимодействие языков. Языковые компиляторы, ориентированные на .NET Framework, выдают промежуточный код, называемый языком CIL (Common Intermediate Language), который в свою очередь компилируется во время выполнения средой CLR. С помощью этой функции подпрограммы, написанные на одном языке, доступны в других языках, поэтому разработчики могут создавать приложения на предпочитаемых языках.

# 8. Лабораторная работа №8. Программы линейной структуры. Класс Math

Задание 1: написать и отладить программу линейной структуры. Ниже приведен пример исходного текста подобной программы.

Составить программу для вычисления значения y(x) и для некоторых x произвести вычисления.

*using System;*

*namespace ConsoleApplication1*

*{ class Class1*

*{ static void Main()*

*{*

*Double x,y; string s; //Описываем переменные*

*Console.Write("Введите x="); //Выдаём приглашение для ввода х*

*s = Console.ReadLine(); //Вводи строку s*

*x = Convert.ToDouble(s); //Преобразовываем строку в х*

*y = (x / 2 \* Math.Exp(x) + Math.Log(1 + Math.Exp(x))) /*

*(Math.Pow(Math.Sin(x), 2)*

*+ Math.Sin(Math.Sqrt(x)));//Вычисляем y согласно формуле*

*Console.Write("y(x)={0}",y); //Печатаем полученный результат*

*Console.ReadKey(); //Ожидаем нажатия на любую клавишу*

*}*

*}*

*}*

# 9. Лабораторная работа №9. Программы ветвящейся структуры. Оператор If

Условный оператор if используется для разветвления процесса вычислений на два направления.

Формат оператора:

if ( логическое\_выражение ) оператор\_1; [ else оператор\_2; ]

Сначала вычисляется логическое выражение. Если оно имеет значение true, выполняется первый оператор, иначе — второй. После этого управление передается на оператор, следующий за условным. Ветвь else может отсутствовать.

Задание 2: написать и отладить программу ветвящейся структуры. Ниже приведен фрагмент исходного текста программы, которая выводит в нисходящем порядке значения трех переменных.

if (x > y)

{

if (x > z)

{

if (y > z) // x - y - z

Console/.WriteLine(“{0},{1},{2}”, x, y, z);

else // x - z - y

Console/.WriteLine(“{0},{1},{2}”, x, z, y);

}

else // z - x - y

Console/.WriteLine(“{0},{1},{2}”, z, x, y);

}

else

{

if (y > z)

{

if (x > z) // y - x - z

Console/.WriteLine(“{0},{1},{2}”, y, x, z);

else // y - z - x

Console/.WriteLine(“{0},{1},{2}”, y, z, x);

}

else // z - y - x

Console/.WriteLine(“{0},{1},{2}”, z, y, x);

}

# 10. Лабораторная работа №10. Оператор switch

Оператор switch (переключатель) предназначен для разветвления процесса вычислений на

несколько направлений.

Формат оператора:

switch ( выражение ){

case константное\_выражение\_1: [ список\_операторов\_1 ]

case константное\_выражение\_2: [ список\_операторов\_2 ]

...

case константное\_выражение\_n: [ список\_операторов\_n ]

[ default: операторы ]

}

Ниже приведен пример программы, реализующей простейший калькулятор на четыре действия.

*using System;*

*namespace ConsoleApplication1*

*{ class Class1*

*{ static void Main()*

*{*

*string buf;*

*double a, b, res;*

*Console.WriteLine( "Введите первый операнд:" );*

*buf = Console.ReadLine();*

*a = double.Parse( buf );*

46

*Console.WriteLine( "Введите знак операции" );*

*char op = (char)Console.Read();*

*Console.ReadLine();*

*Console.WriteLine( "Введите второй операнд:" );*

*buf = Console.ReadLine();*

*b = double.Parse( buf );*

*bool ok = true;*

*switch (op)*

*{*

*case '+' : res = a + b; break;*

*case '-' : res = a - b; break;*

*case '\*' : res = a \* b; break;*

*case '/' : res = a / b; break;*

*default : res = double.NaN; ok = false; break;*

*}*

*if (ok) Console.WriteLine( "Результат: " + res );*

*else Console.WriteLine( "Недопустимая операция" );*

*}*

*}*

*}*

# 11. Лабораторная работа №11. Циклы с известным числом повторений. Итерационные циклы. Сложные циклы

В языке существует 3 оператора цикла:

* оператор цикла с предусловием;
* оператор цикла с постусловием;
* оператор цикла с параметром.

Ниже приведен текст программы табулирования функции

namespace ConsoleApplication1

{ class Class1

{ static void Main()

{

double Xn = -2, Xk = 12, dX = 2, t = 2, y;

Console.WriteLine( "| x | y |" ); // заголовок таблицы

double x = Xn;

while ( x <= Xk )

{

y = t;

if ( x >= 0 && x < 10 ) y = t \* x;

if ( x >= 10 ) y = 2 \* t;

Console.WriteLine( "| {0,6} | {1,6} |", x, y );

x += dX;

}}}

# 12. Лабораторная работа №12. Обработка одномерных массивов. Обработка матриц

Массив — это ограниченная совокупность однотипных величин. Элементы массива имеют одно и то же имя, а различаются по порядковому номеру (индексу).

Массив в С# относится к ссылочным типам данных, то есть располагается в динамической области памяти, поэтому создание массива начинается с выделения памяти под его элементы. Массивы вводятся и выводятся с помощью цикла.

Пример: вычисление суммы отрицательных и количества положительных элементов в заданном массиве.

using System;

namespace ConsoleApplication1

{ class Class1

{ static void Main()

{

const int n = 6;

int[] a = new int[n] { 3, 12, 5, -9, 8, -4 };

Console.WriteLine( "Исходный массив:" );

for ( int i = 0; i < n; ++i )

Console.Write( "\t" + a[i] );

Console.WriteLine();

long sum = 0; // cумма отрицательных элементов

int num = 0; // количество отрицательных элементов

for ( int i = 0; i < n; ++i )

if ( a[i] < 0 )

{

sum += a[i];

++num;

}

Console.WriteLine( "Сумма отрицательных = " + sum );

Console.WriteLine( "Кол-во отрицательных = " + num ); } }}

# 13. Лабораторная работа №13. Обработка строк

Тип string, предназначенный для работы со строками символов в кодировке Unicode,

является встроенным типом C#. Ему соответствует базовый класс System.String

библиотеки .NET.

В классе System.String предусмотрено множество методов, полей и свойств, позволяющих

выполнять со строками практически любые действия.

Пример применения методов приведен ниж.

*using System;*

*namespace ConsoleApplication1*

*{ class Class1*

*{ static void Main()*

*{*

*string s = "прекрасная королева Изольда";*

*Console.WriteLine( s );*

*string sub = s.Substring( 3 ).Remove( 12, 2 ); // 1*

78

*Console.WriteLine( sub );*

*string[] mas = s.Split(' '); // 2*

*string joined = string.Join( "! ", mas );*

*Console.WriteLine( joined );*

*Console.WriteLine( "Введите строку" );*

*string x = Console.ReadLine(); // 3*

*Console.WriteLine( "Вы ввели строку " + x );*

*double a = 12.234;*

*int b = 29;*

*Console.WriteLine( " a = {0,6:C} b = {1,2:X}", a, b ); // 4*

*Console.WriteLine( " a = {0,6:0.##} a = {1,5:0.# ' руб. '}",*

*a, b ); // 5*

*}*

*}*

*}*

Результат работы программы:

прекрасная королева Изольда

красная корова Изольда

прекрасная! королева! Изольда

Введите строку

не хочу!

Вы ввели строку не хочу!

a = 12,23p. b = 1D

# 14. Лабораторная работа №14. Разработка программ с использованием классов

В ходе данной лабораторной работы студент Разработать и описать класс в соответствии с вариантом. Написать приложение, в котором создать несколько экземпляров описанного класса и проверить на них заданные методы и свойства.

Примерный вариант задания следующий:

Создать класс Triangle, разработав следующие элементы класса:

1. Поля: три стороны треугольника.
2. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.
3. Методы, позволяющие:
   * сконвертировать данные в строку;
   * рассчитать периметр треугольника;
4. Свойства:
   * позволяющее получить-установить длины сторон треугольника (доступное для чтения и записи);
   * позволяющее установить, существует ли треугольник с данными длинами сторон (доступное только для чтения).
5. Операции:
   * “-- “: уменьшение максимальной (или одной из максимальных, если их несколько) стороны на единицу;
   * “+” : сумма периметров заданных треугольников.

Исходный текст приложения представлен ниже.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

public class Triangle

{

private float a, b, c;

Boolean res;

public float A

{

set { a = value; }

get { return (a); }

}

public float B

{

set { b = value; }

get { return (b); }

}

public float C

{

set { c = value; }

get { return (c); }

}

public float p(float a, float b, float c)

{

return a + b + c;

}

public string s(float x)

{ return Convert.ToString(x); }

public readonly Boolean Res (float a, float b, float c)

{

if (a < b + c && b < a + c && c < a + b)

res=true;

else res=false;

}

public Triangle(float a, float b, float c)

{

A = a; B = b; C = c;

}

static void Main(string[] args)

{

Triangle t = new Triangle(3, 4, 5);

Console.WriteLine("treug syshesnvuet {0}", t.res);

Console.WriteLine("perimetr= {0}", t.p(t.a, t.b, t.c));

Console.WriteLine("stroka {0}", t.s(t.a));

Console.ReadLine(); }