Министерство общего и профессионального образования Свердловской области

ГАПОУ СО «ЕКАТЕРИНБУРГСКИЙ ЭКОНОМИКО-ТЕХНОЛОГИЧЕ-

СКИЙ КОЛЛЕДЖ»

**ПРАКТИКУМ**

**по учебной дисциплине**

**ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

профессиональный цикл

технический профиль

специальности 09.02.07 «Информационные системы и программирование»

ДЛЯ СТУДЕНТОВ ОЧНОЙ И ЗАОЧНОЙ ФОРМ ОБУЧЕНИЯ

Екатеринбург, 2017 г.

**СОДЕРЖАНИЕ**

Практическая работа №1 «Составление блок-схем алгоритмов. Алгоритмы

линейной структуры» .................................................................................................. 3

Практическая работа № 2 «Алгоритмы разветвляющейся структуры» ................ 9 Практическая работа № 3 «Алгоритмы циклической структуры» ...................... 15 Практическая работа №4 «Консольные и линейные приложения» ..................... 20

Практическая работа №5 «Создание проектов с использованием условного оператора if/else» ....................................................................................................... 42

Практическая работа № 6 «Создание проектов с использованием оператора выбора switch и перечислений» ............................................................................... 51

Практическая работа № 7 «Создание проектов создание циклической структуры

(for)» ............................................................................................................................ 60

Практическая работа № 8 «Создание проектов создание циклической структуры

(while и do)» ............................................................................................................... 71

Практическая работа №9 «Создание проектов для работы с одномерными массивами» ................................................................................................................. 81

Практическая работа №10 «Создание проектов для работы с двумерными

массивами» ................................................................................................................. 87

Практическая работа №11 «Создание простейших методов» .............................. 93 Практическая работа №12 «Создание простейших классов» ............................... 99 Практическая работа №13 «Создание классов с использованием свойств» ..... 111

Практическая работа №14 «Создание классов, иерархически связанных между собой» ....................................................................................................................... 115

Практическая работа №15 «Создание проектов с использованием текстовых

файлов» ..................................................................................................................... 120

Практическая работа №16 «Создание MDI приложений и меню» .................... 127

Практическая работа №17 «Использование компонентов OpenFileDialog и

SaveFileDialog для работы с файлами» ................................................................. 134

**Тема 1«Основы алгоритмизации»**

**Практическая работа №1 «Составление блок-схем алгоритмов. Алгоритмы линейной структуры»**

2 часа

**Учебная цель:**

1. Познакомиться с основными способами представления алгоритмов.
2. Познакомиться с классификацией и структурой алгоритмических языков.
3. Научиться решать задачи с использованием блок-схем.
4. Научиться решать задачи на составление алгоритмов линейной структуры.

**Учебные задачи:**

1. Углубить и систематизировать знания по блок-схемам.
2. Развить алгоритмическое мышление через составление блок-схем.

**Образовательные результаты, заявленные во ФГОС ТОП-50:**

Студент должен уметь:

* Разрабатывать алгоритмы для конкретных задач;
* Использовать программы для графического отображения алгоритмов. знать:
* Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания по созданию блок-схем.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Блок-схемы являются одним из графических способов представления алгоритмов. Блок-схема состоит из блоков, соединенных линиями. Чаще всего используются блоки следующих типов:

* 1. Блок присваивания

Вычислительное действие или последовательность действий (обрабатывает данные и размещает результаты в ячейки памяти с указанным именем)

* 1. Блок проверки условия.

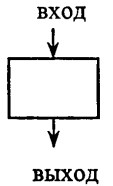
Происходит выбор направления выполнения алгоритма в зависимости от выполнения условия.

* 1. Блок ввод-вывод данных
  2. Блок начала и конца

Может означать начало или конец алгоритма, вход или выход в подпрограмму.

Линейный алгоритм - набор команд, выполняемых последовательно во времени, друг за другом.

Блок-схема базовой структуры следование.



**Вопросы для закрепления теоретического материала:**

1. Что такое линейный алгоритм? Какой схемой он представляется?
2. Каковы основные символы блок-схем?
3. Назовите способы представления алгоритмов.

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Составить простые алгоритмы в двух видах (словесное описание и в виде блок-схемы), а также проанализируйте исходные, результирующие и промежуточные данные с указанием типов данных:

**1.1.** Определить расстояние, пройденное человеком, если известно время, скорость движения, и движение было равномерным (S=v\*t).

Проанализируем данные задачи.

Исходные данные:

V – скорость движения человека, вещественного типа T – время в пути, вещественного типа Результирующие данные:

S – расстояние пройденное человеком, вещественного типа Промежуточных данных в этой задачи нет.

Составим словесный алгоритм.

Ввод V и T

Вычисление s по формуле S=v\*t

Вывод S

Откройте Редактор блок-схем и используя инструменты среды создайте блок-схему задачи, как показано на рисунке 1.

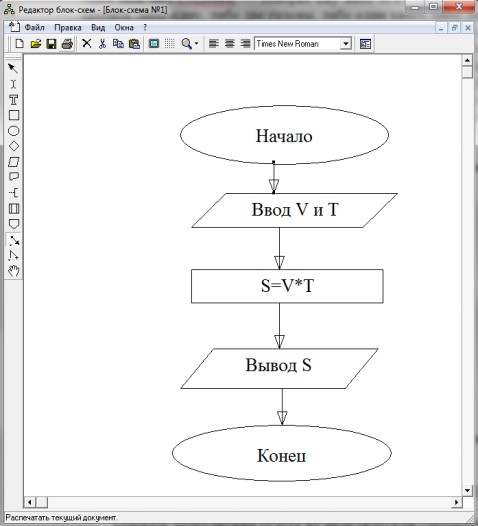


Рисунок 1 - Блок-схема задачи

* 1. Вычислить значение: z = (5+a)/(7-y)
  2. Коля купил у вождя дикого африканского племени 18 кг бананов за 72 блестящие пуговицы. Сколько килограммов бананов сможешь купить у вождя ты, если у тебя на 24 блестящие пуговицы больше, чем у Коли?
  3. Для своего внука дедушка сочиняет в день 4 сказки, это на 1 сказку больше, чем сочиняет бабушка. Дедушка и бабушка сочинили одинаковое количество сказок. Дедушка затратил на сочинение этих сказок 6 дней. Сколько дней сочиняла сказки бабушка?

**Задание 2.** Составьте словесный алгоритм следующей задачи.

* 1. Как перевезти в лодке с левого берега на правый козла, капусту, двух волков и собаку, если известно, что волка нельзя оставлять без присмотра с козлом и собакой, собака в «ссоре» с козлом, а козел «неравнодушен» к капусте? В лодке только три места, поэтому можно брать с собой не более двух животных или одно животное и капусту. Найди несколько способов решения этой задачи.
  2. Собрался Иван-царевич на бой со Змеем Горынычем, трехглавым и треххвостым. «Вот тебе меч-кладенец, — говорит ему Баба Яга. — Одним ударом ты можешь срубить либо одну, либо две головы, либо один хвост, либо два хвоста. Запомни: срубишь голову — новая вырастет, срубишь хвост — два новых вырастут, срубишь два хвоста — голова вырастет, срубишь две головы — ничего не вырастет». За сколько ударов Иван-царевич может срубить Змею все головы и хвосты?
  3. Два мальчика и двое взрослых должны переправиться на другую сторону реки на плоту, который выдерживает либо двух мальчиков, либо одного мальчика и одного взрослого. Как осуществить переправу? Найди несколько способов решения этой задачи.

**Задание 3.** Составить словесное описание, блок-схему, с указанием исходные, результирующие и промежуточные данные задачи по варианту. Номер варианта определяется номером ПК, за которым Вы сидите ил на усмотрение преподавателя. Решите и оформите задачу.

**Вариант 1.** Составьте программу для вычисления периметра прямоугольника.

**Вариант 2.** Составьте программу для вычисления площади трапеции.

**Вариант 3.** Составьте программу для вычисления площади прямоугольного треугольника по его катетам a и b.

**Вариант 4.**  Расстояние между домами, в которых живут Петя и Коля, 1200 м. Однажды они вышли каждый из своего дома и направились навстречу друг другу. Когда Петя прошел А метров, они встретились. Во сколько раз расстояние, которое прошел Коля больше расстояния, которое прошел Петя?

**Вариант 5.** От станции до озера S км. Туристы, направляясь от станции к озеру, полтора часа шли пешком со скоростью А км, а затем сели на попутную машину, которая ехала со скоростью В км/ч. За какое время туристы добрались до озера?

**Вариант 6.** Универмаг в ноябре продал товаров на А млн. р., что на 15% больше, чем в октябре. На какую сумму были проданы товары в октябре?

**Вариант 7.** В классе N учеников. После контрольной было получено: A - пятерок, B - четверок, C -троек. Найти процент троек, четверок и пятерок.

**Вариант 8.** Четырем классам поручено убрать поле площадью F га. Известно количество учеников в каждом классе. Сколько га надо убрать каждому классу?

**Вариант 9.** Вычислить расстояние между двумя точками с данными координатами x1, y1 и x2, y2.

**Вариант 10.** По двум данным катетам найти гипотенузу и площадь прямоугольного треугольника.

**Вариант 11.** Даны числа x и y. Вычислить их сумму, разность и произведение.

**Вариант 12.** Автомобиль проехал три участка пути разной длины с разными скоростями. Найти среднюю скорость автомобиля.

**Вариант 13.** Известна сумма денег, имеющаяся у покупателя и стоимость одной единицы товара. Сколько единиц товара сможет купить покупатель и какова его сдача?

**Вариант 14.** Известен объем продукции, выпускаемый пятью предприятиями отрасли. Вычислить средний объем продукции, выпускаемый одним предприятием.

**Вариант 15.** Известен плановый P и фактический F показатели выпуска продукции предприятием. Определить процент выполнения плана.

**Вариант 16.** Дана сторона равностороннего треугольника. Найти его площадь.

**Вариант 17.** Цех по ремонту бытовой техники имеет ежедневно прибыль Р, с которой платит налог 70%. Из оставшейся доли прибыли 30% идет на развитие производства, а остальное – на заработную плату 8 работникам. Какова среднемесячная зарплата работника?

**Вариант 18.** Расстояние между полицейской машиной и автомобилем преступника равно 240 м. Скорость полицейского автомобиля 40 м/c, машины преступника - 38 м/с. Как скоро полицейские догонят преступника?

**Вариант 19.** Бабушка вяжет в неделю 3 пары детских носков, пару женских и пару мужских и продает их. Считая, что в месяце 4 недели, определить, какую прибыль бабушка имеет за месяц.

**Вариант 20.** За год квартплата повышалась дважды. Первый раз на 30%, а второй на 40%.Год назад семья Березовских платила квартплату 80 рублей в месяц. Сколько они платят сейчас.

**Критерии оценки:**

**«отлично»**

* правильно решены и модифицированы все задания с 1 по 3;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* вариант задания 3 соответствует номеру ПК.

**«хорошо»**

* задания с 1 по 3 решены и модифицированы с ошибками;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* вариант задания 3 соответствует номеру ПК.

**«удовлетворительно»**

* решены и модифицированы задания с 1 по 2 с ошибками;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.

**Порядок выполнения отчета по практической работе**

После завершения работы Вы должны представить отчет, в котором следует:

* указать постановку задачи;
* дать описание исходных, результирующих и промежуточных данных;
* показать словесный алгоритм и блок-схему;
* оформить задачу по варианту, указав номер варианта;
* отчет выполняется на листе формата А4 в печатном виде.

**Образец отчета по практической работе**

**Постановка задачи № 1:** Составить простые алгоритмы в двух видах (словесное описание и в виде блок-схемы), а также проанализируйте исходные, результирующие и промежуточные данные с указанием типов данных определения расстояния, пройденное человеком, если известно время, скорость движения, и движение было равномерным (S=v\*t).

*Исходные данные:*

V – скорость движения человека, вещественного типа

T – время в пути, вещественного типа

*Результирующие данные:*

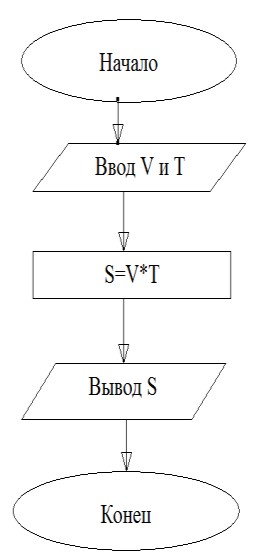
S – расстояние пройденное человеком, вещественного типа

*Промежуточных данных в этой задачи нет*

*Алгоритм*

1. Ввод V и T
2. Вычисление s по формуле S=v\*t
3. Вывод S

*Блок-схема*



**Постановка задачи № 2:** Составьте словесный алгоритм: как перевезти в лодке с левого берега на правый козла, капусту, двух волков и собаку, если известно, что волка нельзя оставлять без присмотра с козлом и собакой, собака в «ссоре» с козлом, а козел «неравнодушен» к капусте? В лодке только три места, поэтому можно брать с собой не более двух животных или одно животное и капусту. Найди несколько способов решения этой задачи. Алгоритм 1.

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. …

**Практическая работа № 2 «Алгоритмы разветвляющейся структуры»**

2 часа

**Учебная цель:**

1. Научиться решать задачи с использованием блок-схем.
2. Научиться использовать базовые структуры ветвление при составлении блок-схем алгоритмов решения задач. **Учебные задачи:**
3. Углубить и систематизировать знаний по блок-схемам.
4. Развить алгоритмическое мышление через составление блок-схем.

**Образовательные результаты, заявленные во ФГОС ТОП-50:**

Студент должен уметь:

* Разрабатывать алгоритмы для конкретных задач;
* Использовать программы для графического отображения алгоритмов. знать:
* Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.

**Краткие теоретические материалы по теме практической работы**

Алгоритм, содержащий хотя бы одно условие, в результате которого обеспечивается переход на один из двух возможных шагов, называется разветвляющимся.

Блок-схема в общем случае отвечает условному оператору и состоит из условия Р и блоков S1 и S2. Если один из блоков отсутствует, то приходим к неполному условному оператору. Она обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран. Структура ветвление существует в следующих основных вариантах:

1. Конструкция ветвления в полной форме (рисунок 2).

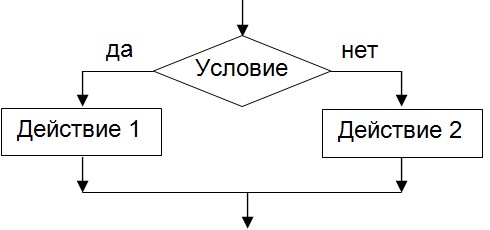


Рисунок 2- Полная форма конструкции ветвления

Если условие истинно, то выполняется действие 1, если условие ложно, то выполняется действие 2.

1. Конструкция ветвления в не полной форме (рисунок 3).

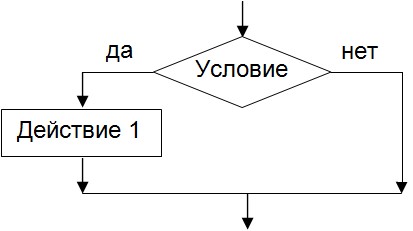


Рисунок 3- Не полная форма конструкции ветвления

Если в ветвлении присутствуют действия только для истинности или только для случая ложности условия, то говорят, что она записана в неполной (в сокращенной) форме. Если выбрано направление, в котором отсутствует действие, то конструкция ветвления не выполняется и управление получает конструкция, следующая за ветвлением.

3.

Выбор

4)

рисунок

(



Условие



Действие 1



Действие 2



Действие 3



Действие 4



Значение 1



Значение 2



Значение 3



Иначе

Рисунок 4- Конструкции выбора

Данная конструкция служит для выбора одного из возможных вариантов в ситуации, которая может иметь несколько (более двух) возможных исходов. В зависимости от значения переменой выполняется оператор, помеченный соответствующей константой. Если же ни одна константа не подходит, выполняется оператор, записанный по ветке иначе.

**Вопросы для закрепления теоретического материала:**

1. Что такое разветвляющий алгоритм и с помощью, каких схем его можно представить?
2. Назовите основные варианты структуры ветвления.

**Задания и инструкция по выполнению практической работы**

**Задание 1.** По данной блок-схеме алгоритма (рисунок 5) вычислите значения Y. Заполните таблицу значений.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | 6 | 0 | -1 | 5 |
| y |  |  |  |  |



Рисунок 5 – Блок-схема алгоритма Заполним таблицу, когда А=6.

* Проверяем условие А=5? 6=5? Нет, следовательно выполняем действие по ветке Нет.
* Проверяем условие А=0? 6=0? Нет, следовательно выполняем действие по ветке Нет.
* Вычисляем выражение н по формуле 2/(А\*(А-5)) , т. е. 2/(6\*(65))=2/(6\*1)=2/6=0,33
* Записываем ответ в таблицу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | 6 | 0 | -1 | 5 |
| y | 0,33 |  |  |  |

* Остальные клетки заполните самостоятельно, выполним аналогичные действия.

**Задание 2.** В блок-схеме алгоритма (рисунок 6), имитирующего работу микрокалькулятора (выполнение 4-х арифметических действий), заполните пустые блоки. К — знак арифметической операции. Перерисуйте блок-схему в редакторе блок-схем, заполнив пустые блоки.

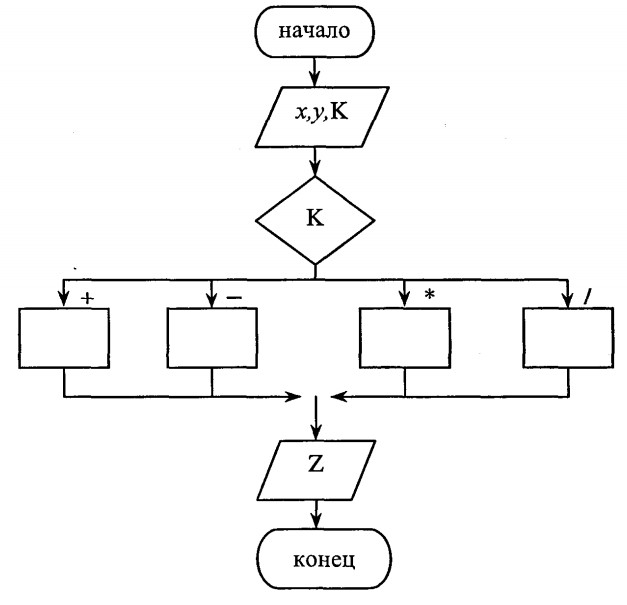


Рисунок 6 – Блок-схема алгоритма

**Задание 3.** Заданы три числа. Известно, что два равны между собой, а третье отлично от них. Заполните шаблон блок-схемы (рисунок 7) нахождения числа, отличного от двух других. Перерисуйте блок-схему в редакторе блок-схем, заполнив пустые блоки.

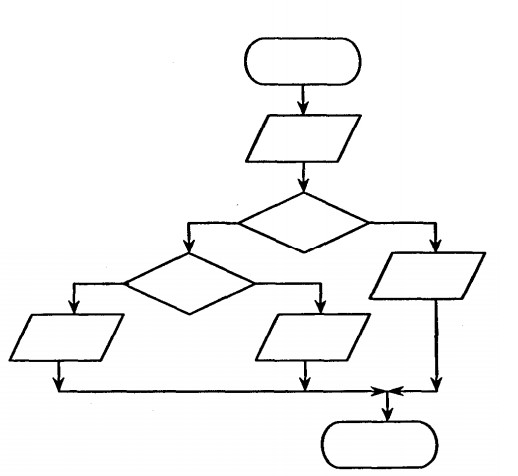
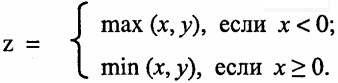


Рисунок 7 – Шаблон блок-схемы алгоритма **Задание 4.** Найти значение



В шаблоне блок-схемы алгоритма расставьте «да» или «нет» и заполните пустые блоки. Перерисуйте блок-схему (рисунок 8) в редакторе блок-схем, заполнив пустые блоки.

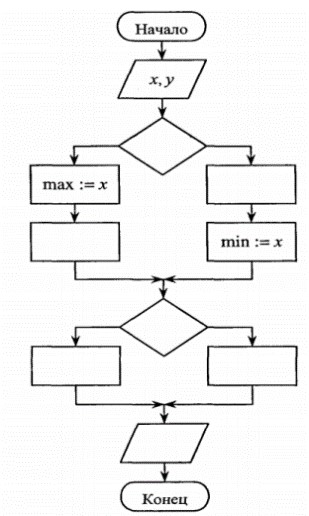


Рисунок 8 – Шаблон блок-схемы алгоритма

**Задание 5.** Составить алгоритмы с использованием схемы ветвления в двух видах (словесное описание и в виде блок-схемы):

5.1. Определения наибольшего из двух заданных целых чисел А и В.

5.2. Определения значение переменной, которая вычисляется по формуле:

y = a + b, если а – нечетное и y = a\*b, если а – четное (условие четности числа: остаток(а/2) = 0).

**Задание 6.** Составить словесное описание, блок-схему, с указанием исходные, результирующие и промежуточные данные задачи по варианту. Номер варианта определяется номером ПК, за которым Вы сидите ил на усмотрение преподавателя. Решите и оформите задачу.

**Вариант 1.** Имеется прямоугольное отверстие со сторонами а и b и кирпич с ребрами х, у, z. Требуется определить, пройдет ли кирпич в отверстие.

**Вариант 2.** Даны два конверта прямоугольной формы с длинами сторон (а,

b) и (с, d). Определить, можно ли один из конвертов вложить в другой.

**Вариант 3.** Дано двузначное число. Написать программу определения:

превышает ли сумма его цифр число, которое вводится с клавиатуры.

**Вариант 4.** Проверить, делится ли введенное трехзначное число нацело на сумму цифр, составляющих это число.

**Вариант 5.** Дано двузначное число. Написать программу, определяющую, какая из цифр этого числа больше.

**Вариант 6.** Заданы координаты двух точек на плоскости. Составьте программу, определяющую, какая из точек удалена от начала координат на большее расстояние. Расстояние между двумя точками на плоскости с координатами (x1,y1) и (x2,y2) определяется по формуле: *S*  (*x*1 *x*2)2 (*y*1 *y*2)2 .

**Вариант 7.** Написать программу, в которой подсчитывается сумма только положительных чисел из трех данных.

**Вариант 8.** Написать программу, в которой подсчитывается количество четных чисел среди трех данных.

**Вариант 9.** Написать программу, которая по номеру дня недели (целому числу от 1 до 7) выдаёт в качестве результата количество уроков в вашей группе в этот день.

**Вариант 10.** Составить программу, которая по заданному году и номеру месяца m определяет количество дней в этом месяце.

**Вариант 11.** Написать программу, позволяющую по последней цифре числа определить последнюю цифру его квадрата.

**Вариант 12.** Для каждой введённой цифры (0-9) вывести соответствующее ей название на английском языке (0-zero, 1-one, 2-two, …).

**Вариант 13.** Составить программу, которая по данному числу (1-12) выводит название соответствующего ему месяца.

**Вариант 14.** Составить программу, позволяющую получит словесное описание школьных отметок (1-“плохо”, 2-“неудовлетворительно”, 3-“удовлетворительно”, 4-“хорошо”, 5-“отлично).

**Вариант 15.** Известны длины двух треугольников. Определить наибольший периметр треугольника.

**Вариант 16.** Написать программу, которая бы по введённому номеру времени года (1- зима, 2 – весна, 3- лето, 4 - осень) выдавала соответствующие этому времени года месяцы, количество дней в каждом из месяцев.

**Вариант 17.** Составить программу, которая определяет, делиться ли введенное с клавиатуры целое число на 7.

**Вариант 18.** Составить программу, которая проверяет, является ли введенное с клавиатуры число одновременно отрицательным и четным.

**Вариант 19.** Напишите программу-модель анализа пожарного датчика в помещении, которая выводит сообщение «Пожарная ситуация», если температура (ее значение вводится с клавиатуры) в комнате превысила 60ºС.

**Вариант 20.** Рис расфасован в два пакета. Вес первого - m кг, второго – n кг. Составьте программу, определяющую: а) какой пакет тяжелее – первый или второй? б) вес более тяжелого пакета.

**Критерии оценки:**

**«отлично»**

* правильно решены и модифицированы все задания с 1 по 6;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* вариант задания 6 соответствует номеру ПК.

**«хорошо»**

* задания с 1 по 6 решены и модифицированы с ошибками;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* вариант задания 6 соответствует номеру ПК.

**«удовлетворительно»**

* решены и модифицированы задания с 1 по 5 с ошибками;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.

**Порядок выполнения отчета по практической работе**

После завершения работы студенты должны представить отчет, в котором следует:

* указать постановку задачи;
* дать описание исходных, результирующих и промежуточных данных;
* показать словесный алгоритм и блок-схему;
* показать заполненные таблицы и блок-схемы;
* оформить задачу по варианту, указав номер варианта;
* отчет выполняется на листе формата А4 в печатном виде.

Образец отчета к данной практической работе имеет тот же вид, что в практической работе № 1.

**Практическая работа № 3 «Алгоритмы циклической структуры»**

2 часа

**Учебная цель:**

1. Научиться решать задачи с использованием блок-схем.
2. Научиться решать задачи на составление алгоритмов циклической структуры.

**Учебные задачи:**

1. Углубить и систематизировать знаний по блок-схемам.
2. Развить алгоритмическое мышление через составление блок-схем.

**Образовательные результаты, заявленные во ФГОС ТОП-50:**

Студент должен уметь:

* Разрабатывать алгоритмы для конкретных задач;
* Использовать программы для графического отображения алгоритмов. знать:
* Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания по созданию блок-схем.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми данными, называется циклическим.

Различают два вида циклов:

1. Циклы с заранее известным числом повторений (Цикл с параметром).

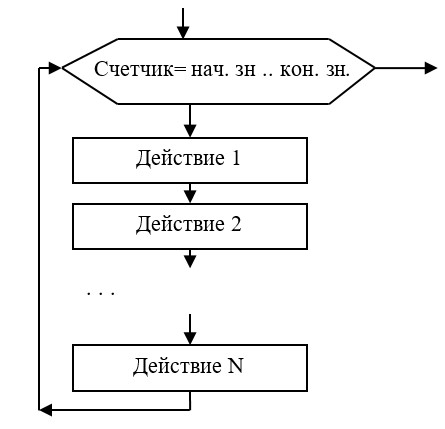


Рисунок 9 –Блок-схемы цикла с параметром

Цикл с параметром (рисунок 9) используется, когда известно начальное значение переменной, конечное значение и шаг изменения равен 1 или –1, т.е. параметр увеличивается или уменьшается на единицу. Таким образом, цикл с параметром организует выполнение одного или нескольких операторов заранее определенное число раз.

1. Циклы с заранее неизвестным числом повторений.

В циклах с заранее неизвестным числом повторений для того, чтобы определить момент прекращения выполнения тела цикла, используется условие цикла. Если при истинности условия цикл продолжается, то такое условие называется условием продолжения цикла.

Если при истинности условия цикл завершается, то такое условие называется условием завершения цикла. В этом случае цикл продолжается до тех пор, пока условие не станет истинным.

Различают два вида циклов с заранее неизвестным числом повторений:

– Циклами с предусловием (рисунок 10).

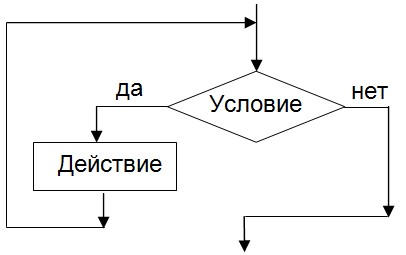


Рисунок 10 –Блок-схемы цикла с предусловием

Сначала проверяется условие. Если оно истинно, то выполняется тело цикла. Если условие становится ложным, то тело цикла не выполняется, а выполняется следующий за оператор за циклом. – Цикл с постусловием (рисунок 11).

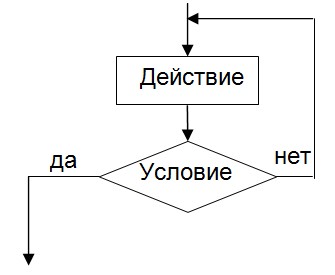


Рисунок 11 –Блок-схемы цикла с постусловием

Сначала выполняется тело цикла, затем проверяется условие. Если оно ложно, то выполняется тело цикла. Если условие истинно, то цикл считается выполненным.

**Вопросы для закрепления теоретического материала:**

1. Что такое циклический алгоритм?
2. Какие виды циклических алгоритмов вы знаете?

**Задания и инструкция по выполнению практической работы**

**Задание 1.** По словесному алгоритму вычисления 2\*n составьте блок-схему алгоритма (n-натуральное число). **Шаг 1.** Задать n

**Шаг 2.** р=1

**Шаг 3.** t=l

**Шаг 4.** Пока i < n то п.5. иначе п. 7

**Шаг 5.** р= р\* 2

**Шаг 6.** i = i + 1, перейти к п. 4

**Шаг 7.** Вывод р

**Шаг 8.** Конец

Откройте редактор блок-схем и составьте блок-схему.

**Задание 2.** Определите значение целочисленной переменной х после выполнения следующего фрагмента блок-схемы (рисунок 12).

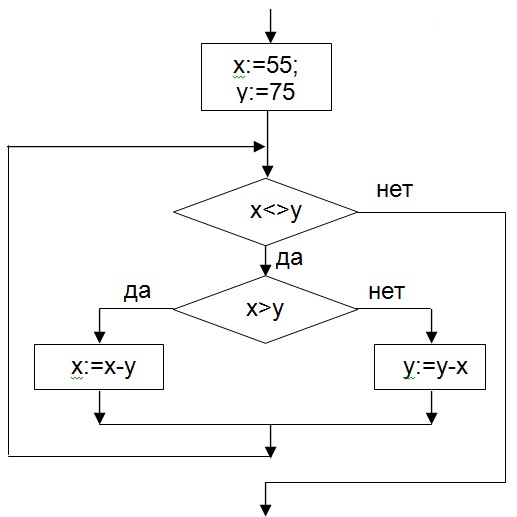


Рисунок 12- Блок-схема задания 2

Для того, чтобы не ошибиться при выполнении блок-схемы, составим, в которую будем заносить значения переменных и результаты проверки условий на каждом шаге.

Таблица 1 – Пошаговове выполнение алгоритма

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № ите-  рации | Значение  х | Значение у | x<>y |  | x>y |
| 0 | 55 | 75 | 55<>75 – да Выполняем цикла | тело | 55>75 – нет, y:=y-x=75-55=20 |
| 1 | 55 | 20 | 55<>20 – да Выполняем цикла | тело | 55>20 – да x:=x-y=55-20=35 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Заполните остальные строки таблицы, выяснив значение х.

**Задание 3.** Составить алгоритмы и блок-схемы для следующих задач:

3.1. Составьте программу, выводящую на экран квадраты чисел от 10 до 20. 3.2. Составить алгоритмы и программы перевода старинных русских мер длины, торгового и аптекарского веса (счетчик цикла меняется от 1 до 10) саженей в метры (1 сажень равна 2,1366 м).

3.3. Бизнесмен взял ссуду m тысяч рублей в банке под 20% годовых. Через сколько лет его долг превысит s тысяч рублей, если за это время он не будет отдавать долг?

**Задание 4.** Составить словесное описание, блок-схему, с указанием исходные, результирующие и промежуточные данные задачи по варианту. Номер варианта определяется номером ПК, за которым Вы сидите ил на усмотрение преподавателя. Решите и оформите задачу.

**Вариант 1.** Начав тренировки, лыжник в первый день пробегал 10 км. Каждый следующий день он увеличивал длину пробега на 10% от предыдущего дня. Определить в какой день он пробежит больше 20 км, в какой день суммарный пробег за все дни превысит 100 км.

**Вариант 2.** Определить средний рост девочек и мальчиков одного класса. Количество учеников в классе заранее неизвестно.

**Вариант 3.** С клавиатуры вводятся N чисел. Составьте программу, которая определяет количество отрицательных, количество положительных и количество нулей среди введенных чисел. Значение N вводится с клавиатуры.

**Вариант 4.** В сберкассу на трехпроцентный вклад положили S рублей. Какой станет сумма вклада через N лет?

**Вариант 5.** Даны натуральные числа от 20 до 50. Напечатать те из них, которые делятся на 3, но не делятся на 5.

**Вариант 6.** Напечатать те из двузначных чисел, которые делятся на 4, но не делятся на 6.

**Вариант 7.** В бригаде, работающей на уборке сена, имеется N сенокосилок. Первая сенокосилка работала m часов, а каждая следующая на 10 минут больше, чем предыдущая. Сколько часов проработала вся бригада?

**Вариант 8.** Задано натуральное число N. Найти количество натуральных чисел, не превосходящих N и не делящихся ни на одно из чисел 2, 3, 5.

**Вариант 9.** У первоклассника Пети m рублей. Мороженое стоит k рублей. Петя решил наесться досыта мороженого, для этого он покупал по одному мороженому и съедал ее до тех пор, пока ему хватало денег. Как Пете узнать, сколько денег останется у него в конце концов? Учтите, что Петя делить еще не умеет, а умеет только вычитать и складывать. Сколько мороженых он может съесть?

**Вариант 10.** Имеется кусок ткани длиной М метров. От него последовательно отрезаются куски разной длины. Все данные по использованию ткани заносятся в компьютер. Компьютер должен выдать сообщение о том, что материала не хватает, если будет затребован кусок ткани, большей длины, чем имеется.

**Вариант 11.** На соревнованиях по фигурному катанию оценки заносятся в компьютер. Составить программу для вывода на экран лучшего результата после каждого выступления фигуристов.

**Вариант 12.** Дано действительное число А, большее 1. Найти среди чисел 1, 1+1/2, 1+1/2+1/3, ... первое, большее А и его номер.

**Вариант 13.** С клавиатуры вводятся числа и последовательно суммируются. Найти количество введенных чисел, когда их сумма превысила 100.Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня.

**Вариант 14.** Найти все двузначные числа, сумма квадратов цифр которых делится на 13.

**Вариант 15.** Какой суммарный путь пробежит спортсмен за 7 дней?

**Вариант 16.** Дано натуральное число n. Составить программу, подсчитывающую количество цифр числа n.

**Вариант 17.** Напишите программу, которая определит первое отрицательное число последовательности: y = sin(i/100), i = 1,2,3…

**Вариант 18.** Составьте программу вычисления суммы всех двузначных чисел.

**Вариант 19.** Дано натуральное n. Вычислить: 1/11 + 1/22 + ... + 1/n n. **Вариант 20.** Найти произведение двузначных нечетных чисел, кратных 13.

**Критерии оценки:**

**«отлично»**

* правильно решены и модифицированы все задания с 1 по 4;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* вариант задания 4 соответствует номеру ПК.

**«хорошо»**

* задания с 1 по 4 решены и модифицированы с ошибками;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* вариант задания 6 соответствует номеру ПК.

**«удовлетворительно»**

* решены и модифицированы задания с 1 по 3 с ошибками;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.

**Порядок выполнения отчета по практической работе**

После завершения работы студенты должны представить отчет, в котором следует:

* указать постановку задачи;
* дать описание исходных, результирующих и промежуточных данных;
* показать словесный алгоритм и блок-схему;
* показать заполненные таблицы и блок-схемы;
* оформить задачу по варианту, указав номер варианта;  отчет выполняется на листе формата А4 в печатном виде.

Образец отчета к данной практической работе имеет тот же вид, что в практической работе № 1.

**Тема 2. «Основы программирования»**

**Практическая работа №4 «Консольные и линейные приложения»**

4 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Научиться использовать элементы кнопка, текстовый блок и метка.
3. Научиться составлять программы для решения задач линейной структуры.
4. Научиться создавать программы линейной структуры.
5. Освоить использование различных способов вывода данных. **Учебные задачи:**
6. Вычислить простейшие математические выражения.
7. Научиться настраивать внешний вид элементов.
8. Научится использовать встроенные математические функции для вычисления выражений.
9. Научиться создавать проекты линейной структуры на С#.

**Образовательные результаты, заявленные во ФГОС ТОП-50:**

Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Понятие «консоль» пришло к нам из тех времен, когда были популярны большие компьютеры, их называли мейнфреймами. Компания размещала в каком-нибудь помещении один гигантский компьютер, а на рабочих местах служащих устанавливалась только клавиатура и простенький монитор, называвшийся консолью. Клавиатура и монитор подключались к тому самому "монстру", спрятанному от посторонних глаз в отдельной комнате. Такие мониторы не умели отображать графику — только текст. Информация передавалась в мейнфрейм при помощи клавиатуры – основного устройства ввода, а консоль – основное устройство вывода – позволяла компьютеру представлять информацию пользователю.

Типы данных представлены в таблице 2.

Таблица 2 – Типы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Ключевое слово | Тип .NET | Диапазон значений |
| Булевский | bool | Boolean | true, false |
| Целые | sbyte | SByte | –128 — 127 |
| byte | Byte | 0 — 255 |
| short | Int16 | –32768 —32767 |
| ushort | UInt16 |  |
| int | Int32 |
| uint | UInt32 |
| long | Int64 |
| ulong | UInt64 |
| Вещественные | float | Single |
| double | Double |
| Символьный | char | Char | U+0000 — U+ffff |
| Строковый | string | String | длина ограничена объемом доступной памяти |
| object | object | Object | Можно хранить все, что угодно |

Переменная — это величина, которая во время работы программы может изменять свое значение. Все переменные, используемые в программе, должны быть описаны.

Для каждой переменной задается ее имя и тип:

int number;

float x, y;

При объявлении можно присвоить переменной начальное значение (инициализировать).

int number = 100;

Можно использовать не только константы, но и выражения — главное, чтобы на момент описания они были вычислимыми, например:

int b = 1, a = 100; int x = b \* a + 25;

Присваивание – это замена старого значения переменной на новое. Старое значение стирается бесследно. Переменная = выражение;

a = b + c; x = 1; x = x + 0.5;

Операции отношения: проверка на равенство (= =), проверка на неравенство

(!=).

Логические операции: && операции И (and), || операции ИЛИ (or),.

Операция ++/-- увеличивает/уменьшает значение переменной на единицу.

Сложное присваивание в C#: x += 0.5; соответствует x = x + 0.5; x \*= 0.5; соответствует x = x \* 0.5; a %= 3;

Методы ввода и вывода:

Console.ReadLine(“Введите пароль” , x) — считывает строку символов, введенную с клавиатуры (или иного устройства ввода).

Console.WriteLine (“ ”, “Ваш пароль” , x) — выводит текст на экран (или иное устройство вывода) начиная с новой строки.

Console.Write — выводит на экран ряд символов без перехода на новую строку.

**Вопросы для закрепления теоретического материала:**

1. Как можно настроить свойства компонента?
2. Назовите основные свойства, настраиваемые для метки и кнопки.
3. Для чего нужен «Обозреватель решений»?
4. Как сохранить проект?
5. Назовите свойство, настраивающее тип шрифта, его начертание и размер. 6 Какими способами можно разместить компонент на форме?
6. Назовите ключевое слово, используемое для описания вещественного числа.
7. С помощью какой функции можно определить квадратный корень?
8. В каком порядке выполняются действия в выражении?
9. Опишите принцип работы оператора присваивания.
10. Для чего нужен MessageBox?
11. Какое свойство (какого компонента?) необходимо настроить, чтобы цвет шрифта всех компонентов на форме по умолчанию стал красным?
12. Назовите операцию нахождения остатка от деления двух чисел.
13. Что такое консоль?

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Создайте консольное приложение **Visual Studio.**

* Откройте **Visual Studio**, далее нажмите на **Начальной странице** пункт меню **Создать проект** и в разделе **Visual C#** выберите **Консольное приложение.**

В нижней части окна введите имя проекта **Консоль1**.

* Укажите папку для сохранения проекта.

**ВАЖНО!** Для каждого проекта должна быть отдельная папка.

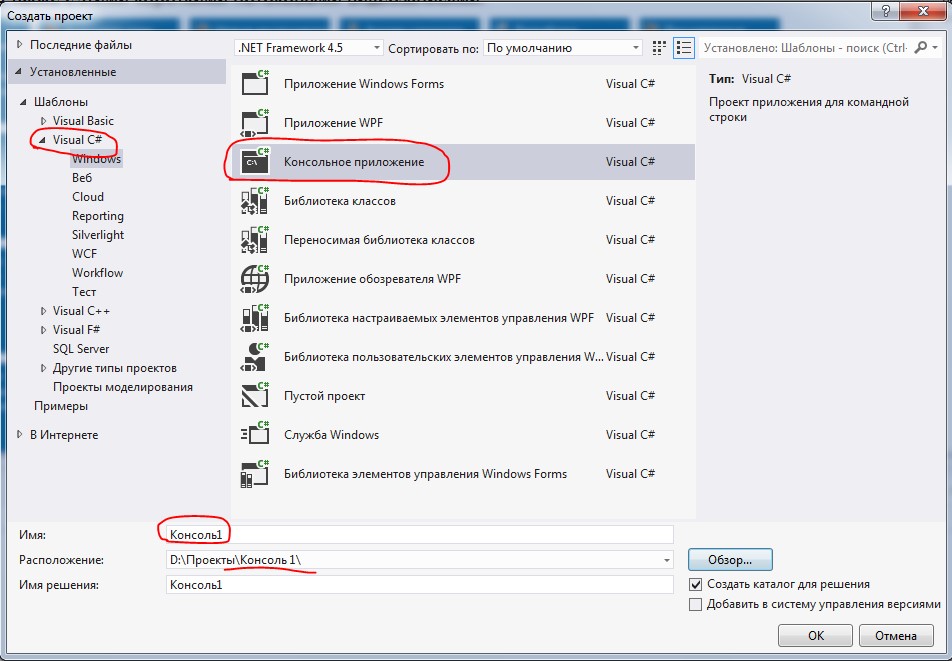


Рисунок 13 – Окно создания проекта

* Далее следует подождать, когда компилятор создаст проект и загрузит консольное приложение.

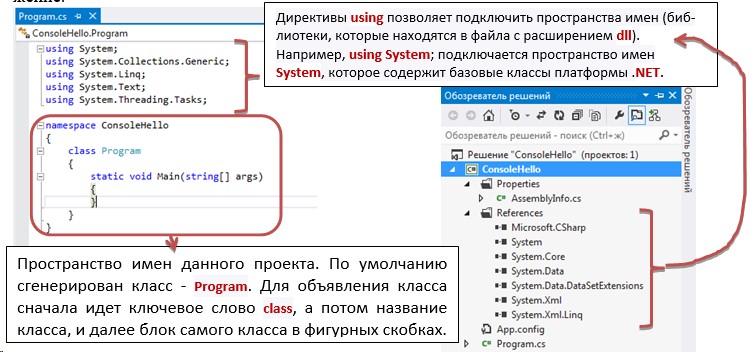


Рисунок 14 – Заготовка программы

Так как C# имеет Си-подобный синтаксис, каждая строка завершается точкой с запятой, а каждый блок кода помещается в фигурные скобки. - Наберите следующий код программы.

Рисунок 15

–

Код программы

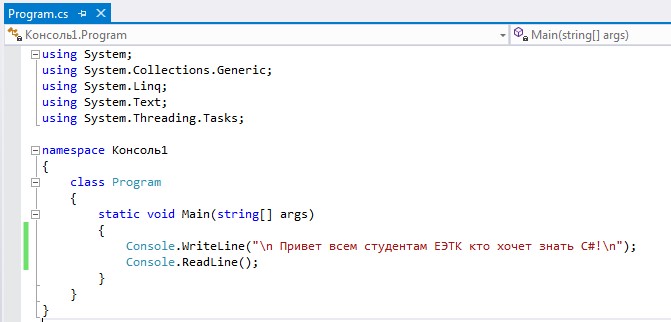
-

Запустите программу на выполнение, нажав на кнопку

или

**F5**

.



Программа выполнит операции.

Текст программы содержит оператор вывода на экран. Вывод осуществляется путем обращения к методу WriteLine. Впереди данной строки и после нее будет пропущено по одной пустой строке, что обеспечивается наличием управляющих символов \n.

Оператор **Console.ReadLine()** позволяет ввести данные, в данном случае ожидает нажатие любой кнопки от пользователя.

**Задание 2.** Напишите программу, в которой описываются и выводятся на экран локальные переменные.

* Создайте консольное приложение **Консоль2.**
* Наберите следующий код программы (рисунок 16).

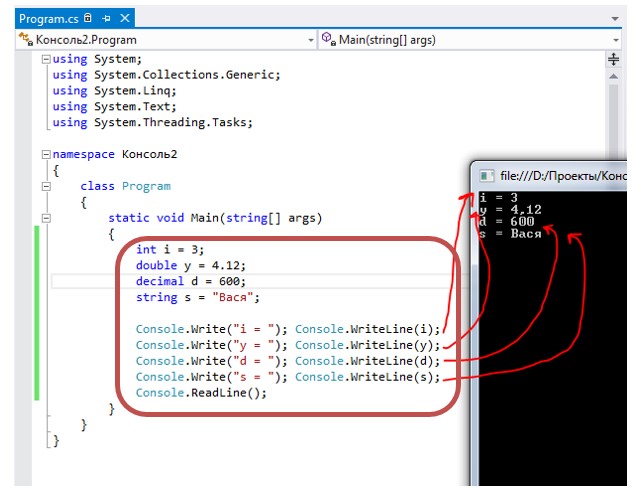


Рисунок 16 – Код программы

**Задание 3.** Напишите программу, которая рассматривает операции ++/--.

* Создайте консольное приложение **Консоль3.**
* Наберите следующий код программы (рисунок 17).

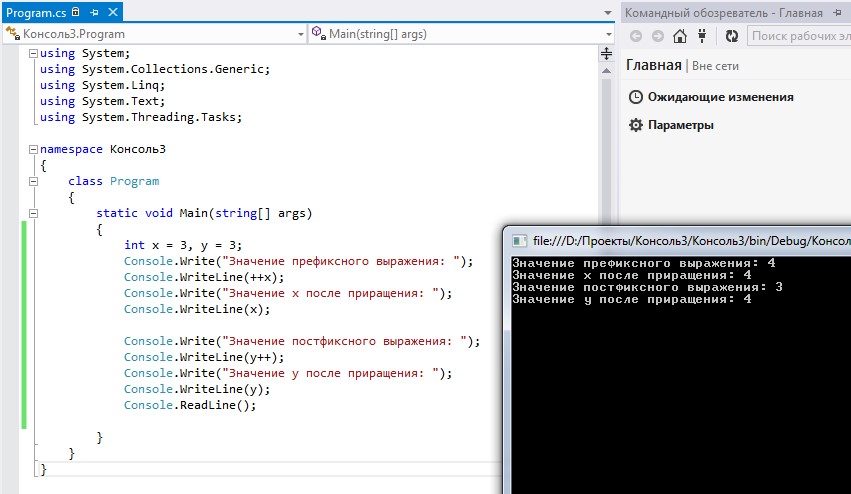


Рисунок 17 – Код программы

**Задание 4.** Напишите программу, которая рассматривает операции ввода/вывода.

* Создайте консольное приложение **Консоль4.**
* Наберите следующий код программы (рисунок 18).

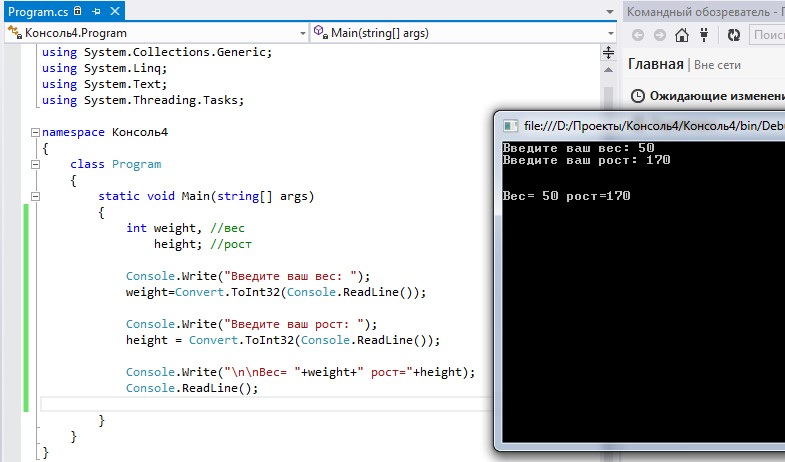


Рисунок 18 – Код программы

В классе Console определены методы ввода строки и отдельного символа, но нет методов, которые позволяют непосредственно считывать с клавиатуры числа. Ввод числовых данных выполняется в два этапа:

1. Символы, представляющие собой число, вводятся с клавиатуры в строковую переменную.
2. Выполняется преобразование из строки в переменную соответствующего типа.

Преобразование можно выполнить с помощью специального класса Convert, определенного в пространстве имен System.

* Строку

Console.Write("\n\nВес= "+weight+" рост="+height); приведите к следующему виду

Console.Write("\n\nВес= "+weight+"\n\nрост="+height);

* Сделайте вывод о назначении \n.

**Задание 5.** Создайте консольное приложение **Консоль5.** Наберите следующий код программы (рисунок 19).

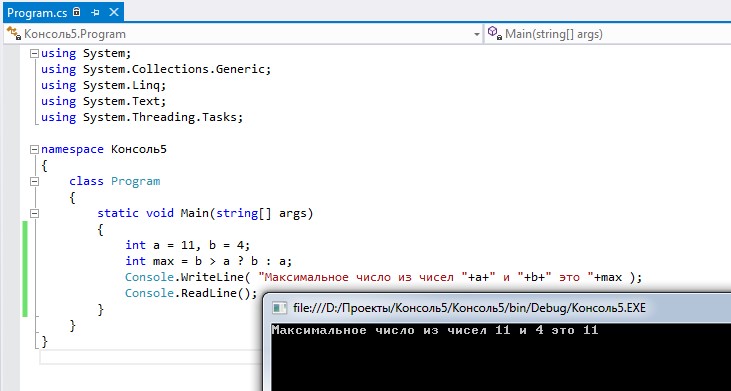


Рисунок 19 – Код программы

В данном проекте требуется вывести в каждой строке не одну, а три величины, поэтому прежде чем передавать их для вывода, их требуется «склеить» в одну строку с помощью операции +.

* Самостоятельно разберитесь в коде программы.

**Задание 6.** Создайте приложение в Visual Studio, позволяющее перевести градусы по Фаренгейте в градусов по Цельсию по формуле

|  |
| --- |
| *C*(*F*32) |

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.

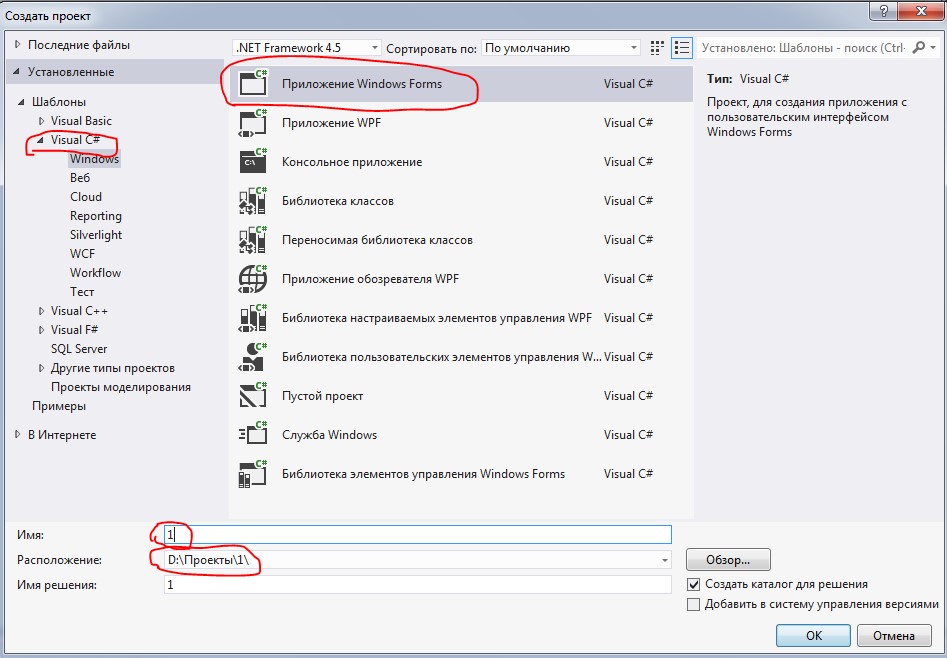


Рисунок 20 – Окно создания проекта

* Далее следует подождать, когда компилятор создаст проект и загрузит главное окно проекта. Microsoft Visual Studio имеет интерфейс, изображенный на рисунке 21.

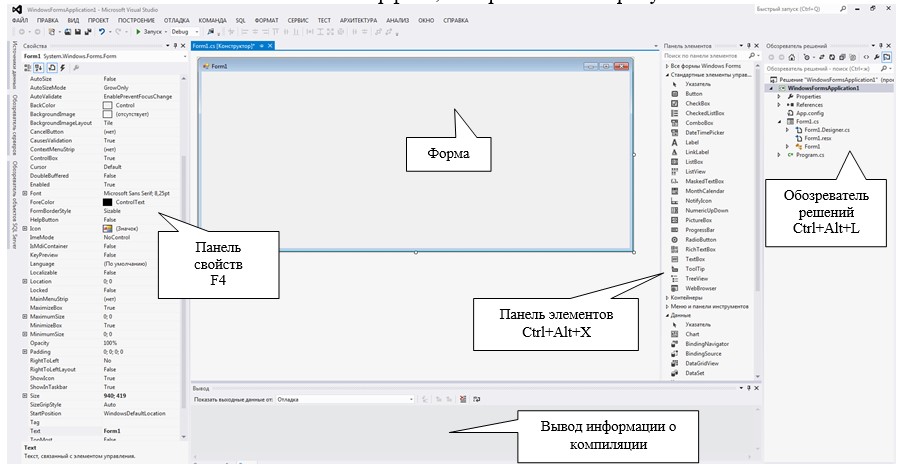


Рисунок 21 – Интерфейс Microsoft Visual Studio

Таблица 3 – Элементы интерфейса Microsoft Visual Studio

|  |  |
| --- | --- |
| Обозреватель реше-  ний(Solution  Explorer) | Позволяет управлять составными частями проекта. |
| Вывод (Output) | Отображает информацию о выполненной компиляции, включая информацию о количестве ошибок |
| Свойства (Properties) | Позволяет настраивать свойства выделенных компонентов |
| Панель элементов (Toolbox) | Отображает имеющиеся элементы |

Разработка проекта приложения в C# состоит из двух взаимосвязанных процессов: конструирование пользовательского интерфейса и написание текста программы, необходимого для решения поставленной задачи.

* Отобразите окно **Обозревателя решений**, для этого в меню **Вид** выберите пункт меню **Обозреватель** **решений** (**Solution** **Explorer**). В данном окне представлена структура построенного решения. Выделите его за заголовок и разместите, как показано на рисунке 22 и нажмите на скрепку для автоматического скрытия окна.

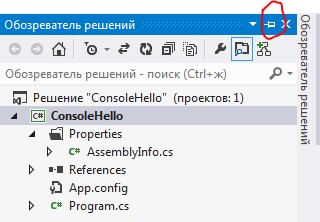
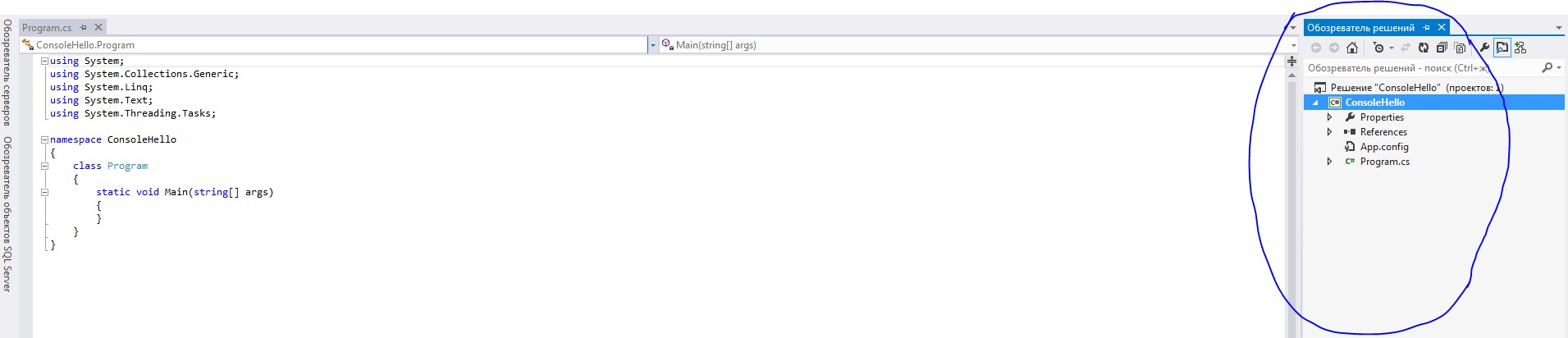


Рисунок 22 – Обозреватель решений

* Отобразите окно **Свойств** (**Properties**), для этого в меню **Вид** выберите пункт меню **Окно** **свойств** или нажмите **F4**. В данном окне можно увидеть свойства выбранного элемента решения. Выделите его за заголовок и разместите, как показано на рисунке 23.

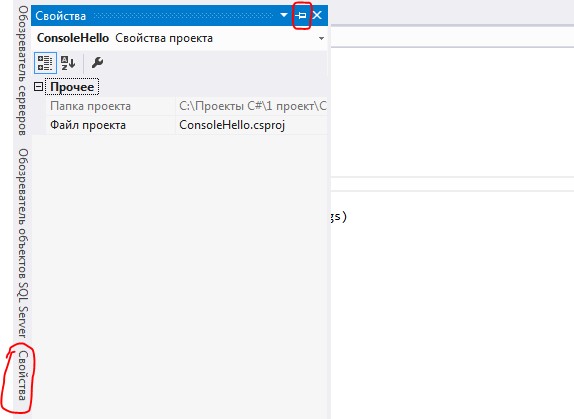


Рисунок 23 – Прикрепление окна к области

* Откройте папку, содержащую проект и Вы увидите следующую файловую структуру (рисунок 24). Заметьте, что мы еще не написали ни одной строчки кода, а в проект уже входит множество файлов, поэтому следует для каждого нового проекта создавать отдельную папку.



Рисунок 24 – Файловая структура проекта

Таблица 4 – Назначение файлов проекта

|  |  |
| --- | --- |
| **Файл** | **Описание** |
| 1.sln | Файл организует проекты, элементы проектов и решений в одно решение. |
| 1.v11.suo | Файл пользовательской информации (User Options File). Отслеживает настройки на уровне пользователя в Visual Studio, такие как контрольные точки. |
| App.config | Файл, используемый для настройки параметров приложения. |
| 1.csproj | Файл проекта Visual Studio. |
| Program.cs | Файл содержит точку входа приложения. |
| Form1.cs | Форма приложений Windows. |
| Form1.Designer.cs | Файл содержит код, автоматически генерированный визуальным конструктором при добавлении элементов управления на форму и изменения их свойств. |
| Form1.resx | Файл, используемый для определения и изменения ресурсов приложения. |

* В Строке заголовка формы укажите название окна проекта (рисунок 25). Для этого в окне **Свойства** в свойстве **Text** для формы, запишите **Перевод градусов**.

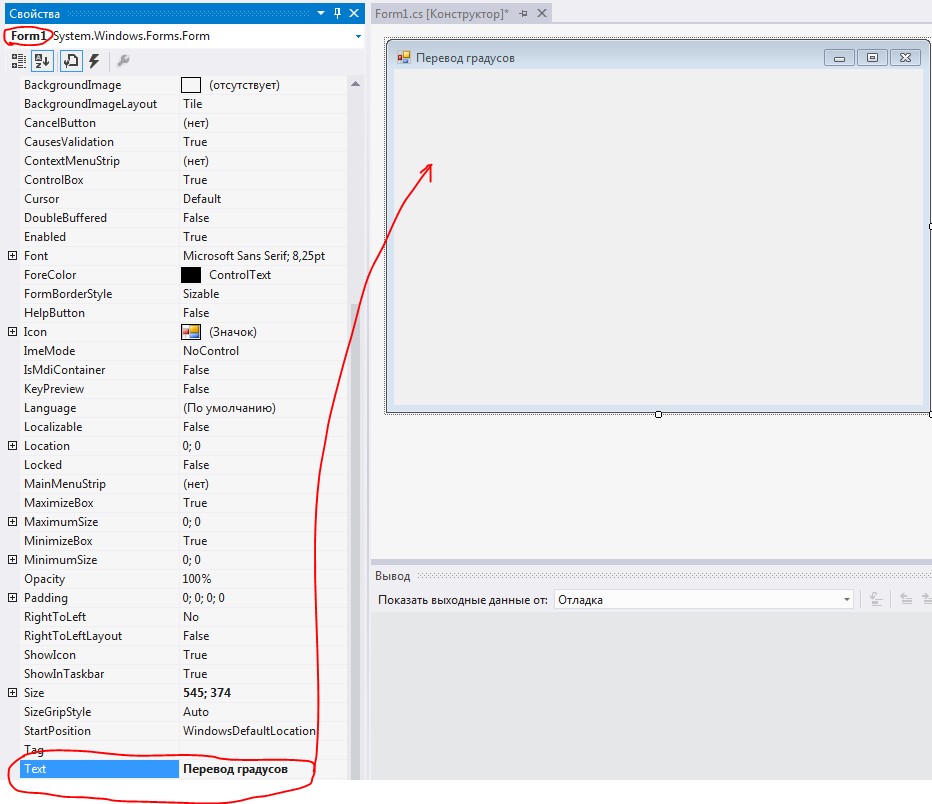


Рисунок 25 – Настройка свойства текст для формы

* На **Панели элементов** в разделе **Стандартные элементы управления** выберите компонент **Button** (кнопка) и щелкните в любом месте формы. При этом на форме появится кнопка, и в Инспекторе объектов будут отображаться ее свойства. В свойстве **Text** для кнопки запишите **Перевести**. В свойстве **Font** увеличьте размер шрифта.

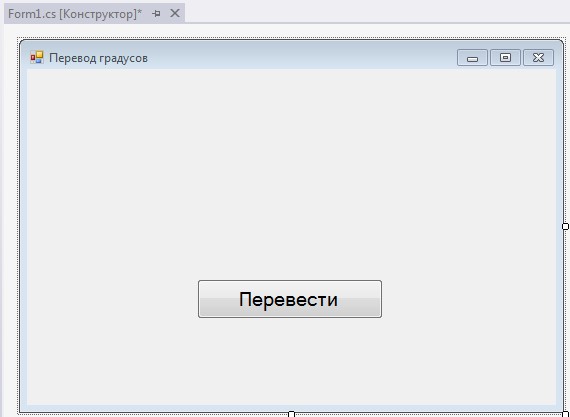


Рисунок 26 – Размещение кнопки на форме

* Исходными данными в этой задачи будут градусы по Фаренгейте. Для ввода этих данных используйте элемент **TextBox**, а для пояснения к полю элемент **Label**. Разместите данные элементы на форме, как изображено на рисунке 27.

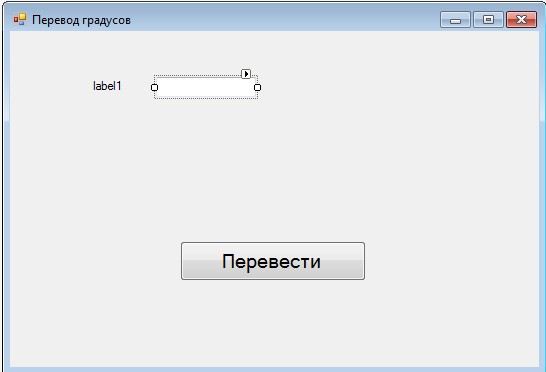


Рисунок 27 – Размещение компонентов на форме прокта - Измените свойства элемента **Label**:

**Text** запишите **Введите градусы по Фаренгейту**.

**Font** измените размер шрифта **14** пт.

* Измените размер шрифта для элемента **TextBox**, с помощью свойства **Font.**

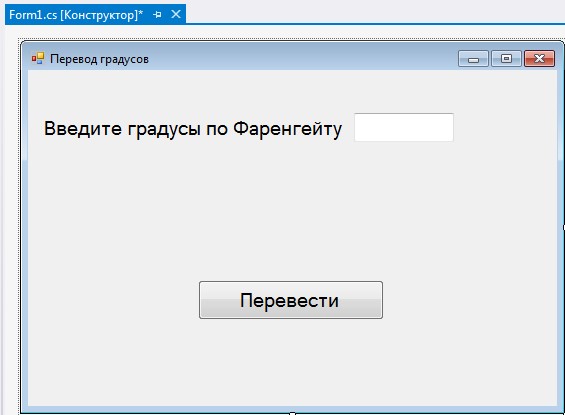


Рисунок 28 – Интерфейс проекта

* Результат расчетов будет отображаться в текстовой строке, поэтому поместите на форму **Label** и в свойство **Text** оставьте пустым, размер шрифта **14** пт**.**

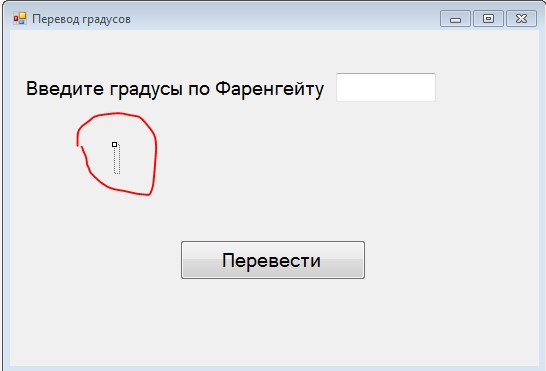


Рисунок 29 – Интерфейс проекта

* Для того чтобы при щелчке на кнопке **Перевести** происходил расчет градусов по Цельсию нужно написать код, обрабатывающий событие «щелчок на кнопке». Дважды щелкните по кнопке **Перевести** левой кнопкой мыши.

При этом откроется программная часть проекта, в которой добавилась следующая запись (рисунок 30):

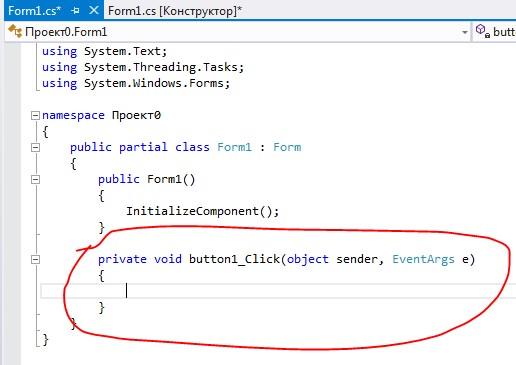
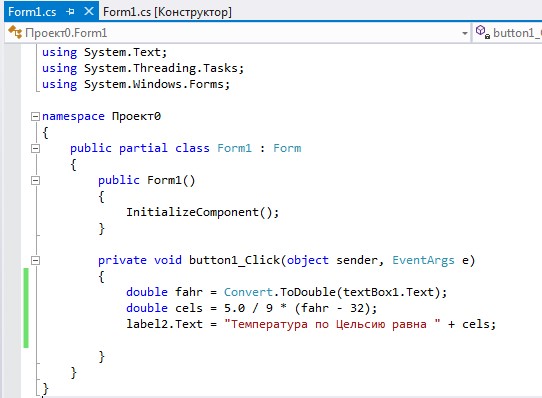


Рисунок 30 – Заготовка процедуры обработки события Click

Это заготовка процедуры обработки события **Click** (щелчок).

* Между **{** (начало) и **}** (конец) надо записать текст процедуры, состоящий из операторов расчета (рисунок 31):



Объявляется переменная

**fahr**

веществен-

ного типа, значение которой берется из эле-

мента на форме

**TextBox1**

из свойства

**Text**

,

предварительно переведя данные в веще-

ственный тип, т. к. в свойстве

**Text**

значения

находятся в типе

**String**

.

Объявляется

переменная

**cels**

вещественного типа,

вычисляющая по формуле

Вывод результата в элемент

на форме

**Label2**

в свойство

**Text**

Рисунок 31 – Код программы

* Запустите приложение на выполнение и протестируйте её (рисунок 32).

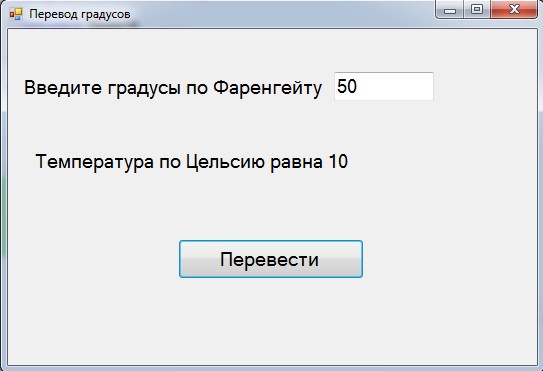


Рисунок 32-Результат тестирования программы

**Задание 7.** Напишите программу «Кнопка». На форме создаются две кнопки, щелчок указателя мыши на которых открывает надпись на форме или удаляет надпись с формы.

* Создайте новый проект.
* В Строке заголовка формы укажите название окна проекта. Для этого в Инспекторе объектов, в свойстве **Text** для формы, запишите: **Кнопка**.
* На **Панели элементов** в разделе **Стандартные элементы управления** выберите компонент **Button** и щелкните в любом месте формы. При этом на форме появится кнопка, и в Инспекторе объектов будут отображаться ее свойства. В свойстве **Text** для кнопки запишите: **Кнопка** **1**. Аналогично добавьте вторую кнопку, только подпись на этой кнопке запишите: **Кнопка 2**. Настройте свойство **Font** для этих кнопок, увеличив размер шрифта **14** пт.

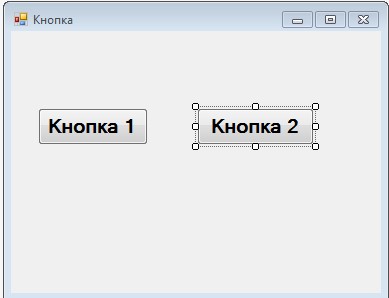


Рисунок 33-Интерфейс приложения

* На **Панели элементов** в разделе **Стандартные элементы управления** выберите компонент **Label**. Щелкните на поле формы, и на нем появится объект **Label,** а Инспектор объектов отобразит его свойства. Свойство **Visible** (видимый) для объекта **Label 1** надо установить в **False** (ложь). Это значит, что текст не будет видимым. В свойстве **Text** запишите: **Привет!**. Выберите параметры для шрифта с помощью свойства **Font** (шрифт) и **ForeColor** (цвет текста).

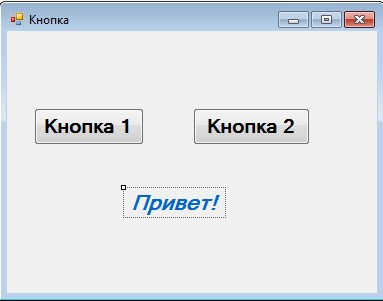


Рисунок 34- Настройка свойств компонентов на форме

* Запустите проект. Убедитесь, что в окне программы есть только кнопки (рисунок 35). Закройте окно программы.



Рисунок 35 – Первый запуск формы

* Для того чтобы при щелчке на кнопке появлялась надпись Привет!, надо написать код, обрабатывающий событие «щелчок на кнопке». Двойным щелчком левой кнопкой мыши щелкните по элементу **Кнопка 1**. При этом откроется программная часть проекта, в которой добавилась следующая запись (рисунок 36):



Рисунок 36- Заготовка процедуры обработки события Click Это заготовка процедуры обработки события **Click** (щелчок).

* Между фигурными скобками следует записать текст процедуры, состоящий из оператора (рисунок 37):

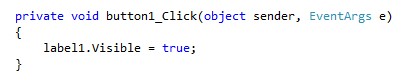


Рисунок 37-Текст процедуры

Этот оператор присваивает свойству **Visible** (видимый) метки **Label1**, находящейся на форме **Form1,** значение **True** (истина). - Запустите программу и убедитесь, что она работает.

**Самостоятельное задание №1.**

Напишите событие **OnClick** для второй кнопки, позволяющий скрывать надпись на форме. Сделайте так, чтобы щелчок на второй кнопке не только удалял надпись «Привет!», но и отображал какую-либо другую надпись, например: «Пока!», другого цвета. Используйте для этого уже имеющуюся метку.

**Задание 8.** Напишите программу «Щелчок на кнопке». При щелчке по различным кнопкам на форме в специальном окне появляется соответствующий текст выбранного цвета.

* Создайте новый проект.
* В Строке заголовка формы укажите название окна проекта. Для этого в Инспекторе объектов, в свойстве **Text** для формы, запишите: **Щелчок**.
* На **Панели элементов** в разделе **Контейнеры** выберите компонент **Panel** и щелкните в любом месте формы (рисунок 38). В свойстве **Text** для панели запишите: **Кнопка** **1**. Аналогично добавьте вторую кнопку, только подпись на этой кнопке запишите: **Кнопка 2**. Настройте свойство **Font** для этих кнопок, увеличив размер шрифта **14** пт.



Рисунок 38- Размещение панели на форме

* На **Панели элементов** в разделе **Стандартные элементы управления** выберите компонент **Button** и разместите в панели четыре кнопки, в свойстве **Text** для этих кнопок запишите: «**Кнопка 1**», «**Кнопка** **2**», «**Кнопка 3**», «**Кнопка 4**». В свойстве **Cursor** (курсор) для каждой кнопки выберите из списка значение: **Hand**. Тогда при наведении курсора на кнопку он приобретет вид ладошки. Настройте свойство **Font** для этих кнопок, увеличив размер шрифта **14** пт и сделав шрифт полужирным (рисунок 39).



Рисунок 39 – Интерфейс приложения

* Под панелью с кнопками вставьте компонент **TextBox**. Настройте свойство **Font** для этих кнопок, увеличив размер шрифта **14** пт (рисунок 40).

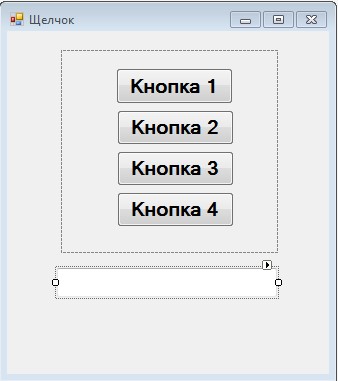


Рисунок 40 – Размещение TextBox на форме

* Для того чтобы при щелчке на кнопке появлялась надпись Привет!, надо написать код, обрабатывающий событие «щелчок на кнопке». Теперь сделайте так, чтобы при щелчке на кнопках, в окне **TextBox1** появлялся текст: «Щелчок на кнопке 1», «Щелчок на кнопке 2», «Щелчок на кнопке 3», «Щелчок на кнопке 4». Двойным щелчком левой кнопкой мыши щелкните по элементу **Кнопка 1**. При этом откроется программная часть проекта (рисунок 41), в которой следует добавить следующие операторы: оператор присваивания свойству **Text** значения, оператор изменения цвета, который берется из системной палитры цветов.



Рисунок 41 – Текст процедуры нажатие на Кнопки 1

* Создайте по аналогии процедуры, обрабатывающие щелчок указателем мыши на других кнопках (рисунок 42). НЕ ЗАБЫВАЙТЕ, что следует вызвать для каждой кнопки событие нажатие кнопки и записать только операторы, которые изображены в фигурных скобках.

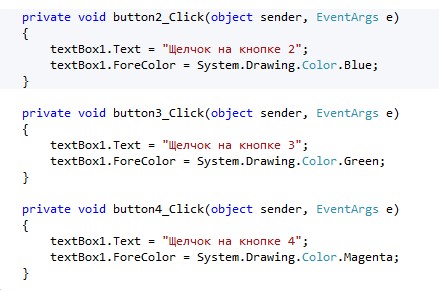


Рисунок 42 – Текст процедур нажатия на кнопки 2-4

* Запустите программу и убедитесь, что она работает (рисунок 43).

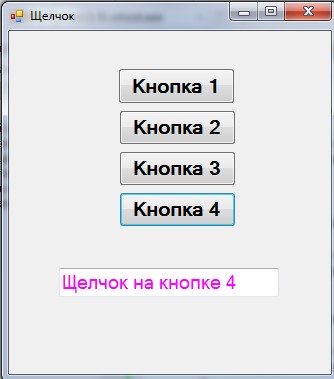


Рисунок 43 – Результат тестирование проекта **Самостоятельное задание №2.**

Добавьте операторы в процедуры для кнопок для того, чтобы при щелчке на каждой кнопке изменялся цвет текста на кнопках.

**Задание 9.** Разработать приложение для управления свойствами кнопок вовремя выполнения приложения.

* Создайте новый проект.
* В Строке заголовка формы укажите название окна проекта. Для этого в Инспекторе объектов, в свойстве **Text** для формы, запишите: **Управление свойствами кнопок**.
* С **Панели элементов** в разделе **Стандартные элементы управления** выберите компонент **Button** и разместите на форме 6 кнопок **Button** и один **Label**. Выровнять их положение и размеры, в соответствии с рисунком 44.

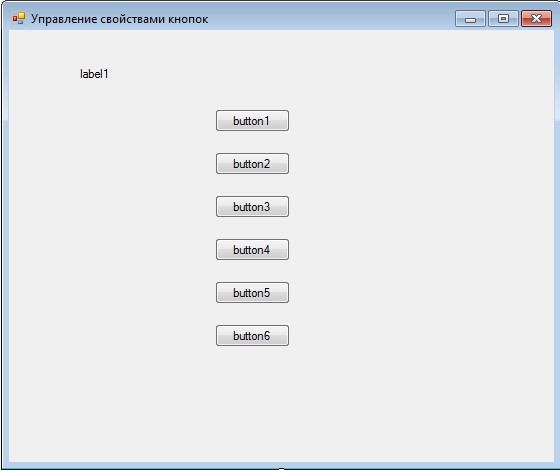


Рисунок 44 – Размещение компонентов на форме

Для выравнивания кнопок следует выделить все кнопки и выровнять их размеры и положение с помощью кнопок .

* Настройте свойства всех элементов помещенных на форму в соответствии с таблицей 5.

Таблица 5 – Значение свойств компонентов проекта

|  |  |  |
| --- | --- | --- |
| Элемент | Свойство | Значение свойства |
| Button1 | Text | Вывести текст на метку |
| Font | размер 14 пт |
| Button2 | Text | Стереть текст с метки |
| Font | размер 14 пт |
| Button3 | Text | Сделать кнопку недоступной |
| Font | размер 14 пт |
| Button4 | Text | Сделать кнопку доступной |
| Font | размер 14 пт |
| Button5 | Text | Сделать кнопку невидимой |
| Font | размер 14 пт |
| Button6 | Text | Сделать кнопку видимой |
| Font | размер 14 пт |
| Label1 | Text | Место для вывода текста |
|  | Font | размер 14 пт, начертание = полужирный |

* После размещения компонентов форма примет вид, приведенный на рисунке 45.

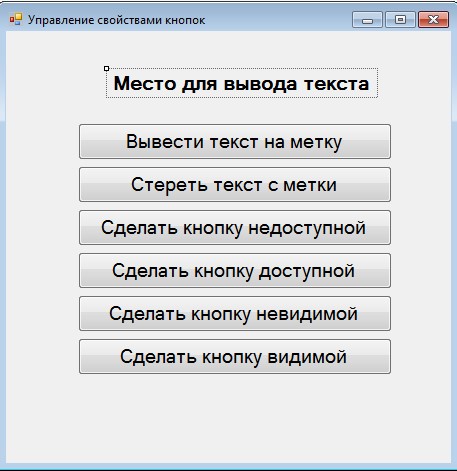
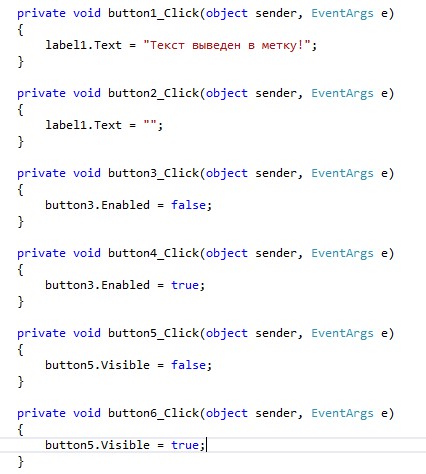


Рисунок 45-Интерфейс приложения

* Напишите события для каждой кнопки (рисунок 46). Не забудьте формировать событие автоматически (двойным щелчком мыши по элементу)



Свойство

**Enabled**

отвечает

за

доступность

Свойство

**Visible**

отвеча

ет за

видимость

эле-

Рисунок 46 – Текст процедур нажатия на кнопки 1-6

* Запустите приложение и проверьте работоспособность всех кнопок формы (рисунок 47).

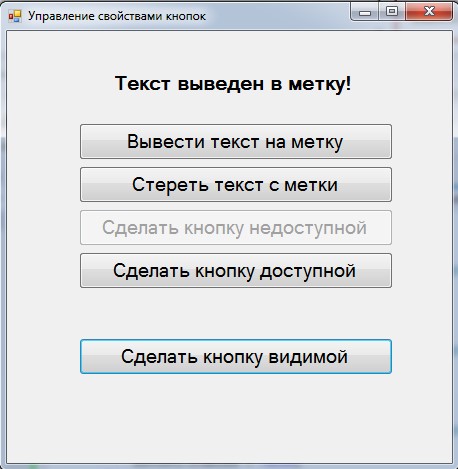


Рисунок 47 – Результат тестирование приложения

**Задание 10.** Создайте приложение в Visual Studio, позволяющее определить количество полных метров в пути, данном в сантиметрах.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Разместите компоненты в соответствии с рисунком 48, настроив свойства данных элементов.

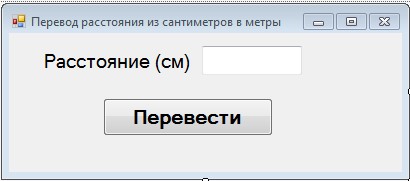


Рисунок 48 – Интерфейс приложения - Создайте событие нажатие кнопки **Перевести**.

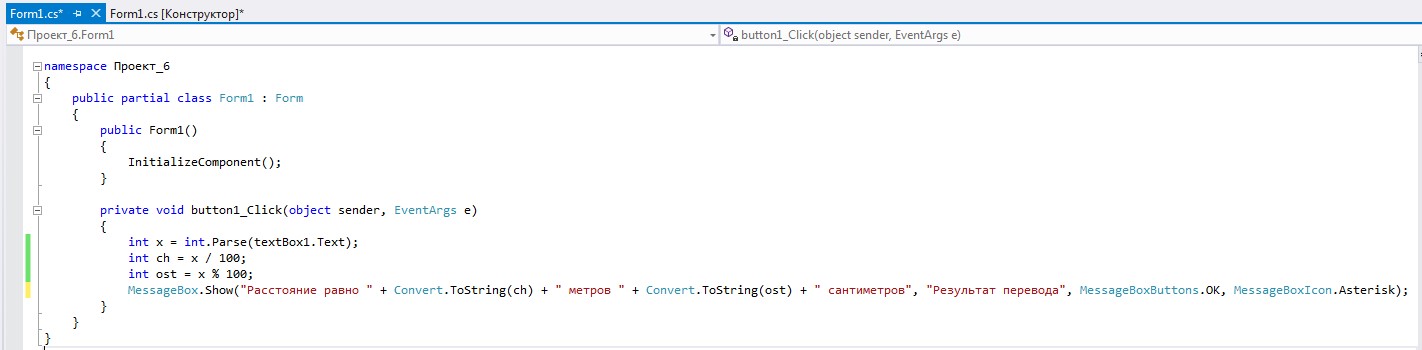


Рисунок 49 – Текст процедуры нажатия на кнопку Перевести Выводить полученные результаты удобно не только с помощью компонентов **label** и **textBox**, но и с использованием объекта **MessageBox**. MessageBox.Show("Текст сообщения", "Заголовок сообщения", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

* Запустите приложение на выполнение и протестируйте программу (рисунок 50).

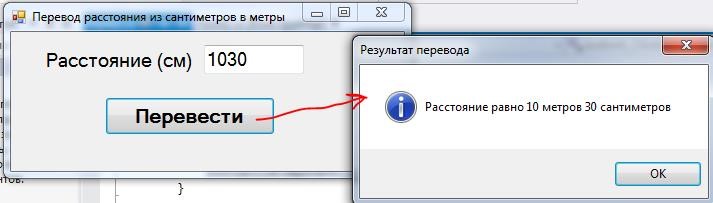


Рисунок 50 – Результат тестирования приложения

**Самостоятельное задание №3.** Напишите программу решения задачи. Исходные данные определите самостоятельно.

1. **вариант.** Даны три целых числа. Найти их сумму и произведение.
2. **вариант.** Дана сторона квадрата a. Найти его периметр и площадь.
3. **вариант.** Дана длина ребра куба a. Найти объем куба и площадь его полной поверхности.
4. **вариант.** Даны три числа. Найти их среднее арифметическое.
5. **вариант.** Известно, что *X* кг конфет стоит *A* рублей. Определить, сколько стоит *Y* кг этих же конфет.
6. **вариант.** Дан размер файла в байтах. Переведите эту величину в килобайты.
7. **вариант.** Дано двузначное число. Найдите сумму и произведение его цифр.
8. **вариант.** С начала суток прошло N секунд. Определите сколько полных часов прошло с начала суток.
9. **вариант.** С начала суток прошло N секунд. Определите сколько полных минут прошло с начала очередного часа;
10. **вариант.** Составьте программу для вычисления периметра прямоугольника.
11. **вариант.** Универмаг в ноябре продал товаров на А млн. р., что на 15% больше, чем в октябре. На какую сумму были проданы товары в октябре?
12. **вариант.** В классе N учеников. После контрольной было получено: A - пятерок, B - четверок, C -троек. Найти процент троек, четверок и пятерок.
13. **вариант.** По двум данным катетам найти гипотенузу и площадь прямоугольного треугольника.
14. **вариант.** Автомобиль проехал три участка пути разной длины с разными скоростями. Найти среднюю скорость автомобиля.
15. **вариант.** Известна сумма денег, имеющаяся у покупателя и стоимость одной единицы товара. Сколько единиц товара сможет купить покупатель и какова его сдача?
16. **вариант.** Известен плановый P и фактический F показатели выпуска продукции предприятием. Определить процент выполнения плана.
17. **вариант.** За год квартплата повышалась дважды. Первый раз на 30%, а второй на 40%.Год назад семья Березовских платила квартплату 80 рублей в месяц. Сколько они платят сейчас.
18. **вариант.** Определить, сколько целых часов (H) и целых минут (М) прошло с начала суток. Например, если k= 13257 = 3 \* 36000 + 40 \* 60 + 57, то H= 3, М= 40. Вывести на экран фразу: «Это...часов...минут».
19. **вариант.** Составить программу вычисления объема и площади поверхности куба по данной длине ребра (V=a3 S=6a2).

**Критерии оценки:**

**«отлично»**

* правильно решены все задания с 1 по 10;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* варианты самостоятельного задания решены верно, самостоятельно.

**«хорошо»**

* задания с 1 по 10 решены с ошибками;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решен только вариант самостоятельного задания № 3.

**«удовлетворительно»**

* решены задания с 1 по 10 с ошибками;
* решены самостоятельные варианты № 1, 2.

**Образец отчета по практической работе**

**Отчет по практической работе №**

Название практической работы: «Консольные и линейные приложения»

Вариант №0

Цели работы:

1. Познакомиться с интегрированной средой разработки Microsoft

Visual Studio.

1. Научиться использовать элементы кнопка, текстовый блок и метка.
2. Научиться составлять программы для решения задач линейной структуры.
3. Научиться создавать программы линейной структуры.
4. Освоить использование различных способов вывода данных.

Внешний вид:

Код программы:

Тест:

Вывод: В ходе выполнения практической работы № я научился ..., освоил ...

.

**Практическая работа №5 «Создание проектов с использованием условного оператора if/else»**

4 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Научиться использовать оператор if при решении задач.
3. Научиться использовать элементы PictureBox, OpenFileDialog.
4. Освоить создание программ разветвляющейся структуры.
5. Освоить использование тернарного оператора и оператора if.
6. Освоить правила записи условных выражений.

**Учебные задачи:**

1. Научиться записывать условия различной степени сложности; 2. Научиться настраивать внешний вид элементов.

3. Создавать проекты разветвляющейся структуры на С# с использованием изученных компонентов.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Синтаксис условного оператора if

**If (**условие**)** оператор; **else** оператор**;**

**if** **(**условие**)** **{**оператор1**;** оператор2**;** операторN**;}** else {оператор1**;** оператор2**;** операторN**;}**

Если значение условия «истинно», то выполняется оператор (им может быть составной оператор - блок), следующий за условием. Если же условие принимает значение «ложно» то выполняется оператор, следующий за ключевым словом **else**. В записи оператора **if** вторая часть (т. е. оператор **else**) может отсутствовать. Тогда, если условие принимает значение «ложно», выполняется сразу следующий оператор программы. В качестве условия может стоять произвольное выражение. В операторе **if** лишь проверяется, является ли значение этого выражения ненулевым (истинным) или нулевым (ложным).

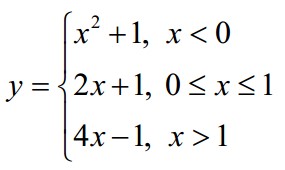
Любая последовательность операторов, заключенная в фигурные скобки, является составным оператором. Составной оператор не должен заканчиваться (**;**), поскольку ограничителем блока служит сама закрывающаяся скобка. Внутри блока каждый оператор должен ограничиваться (**;**).

**Вопросы для закрепления теоретического материала:**

1. Какое условие нужно записать для проверки числа на четность?
2. Перечислите основные логические операции, используемые при записи условий.
3. Объясните разницу между обычными и сокращенными логическими операциями: & и &&, | и ||.
4. Назовите операторы сравнения.

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Составить программу вычисления выражения



* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Разместите компоненты, в соответствии с рисунком 51, настроив свойства данных элементов.

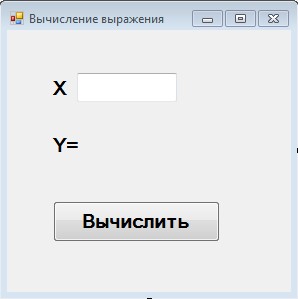


Рисунок 51 – Интерфейс приложения

* Создайте событие нажатие кнопки **Вычислить** (рисунок 52).

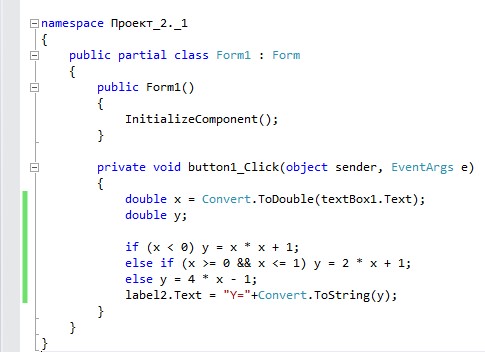


Рисунок 52 – Текст процедуры нажатия на кнопку Вычислить - Запустите приложение на выполнение и протестируйте программу (рисунок 53).

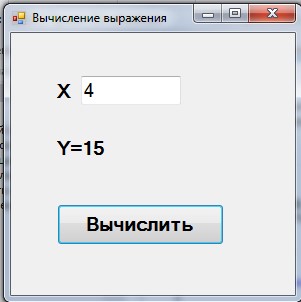


Рисунок 53 – Результат тестирования приложения

**Самостоятельное задание №1**. Создайте новый проект для вычисления значения функции f(x) по варианту (таблица 6).

Таблица 6 – Варианты заданий

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | 13 | 𝑥𝑥, если 𝑥  𝑥 − 25, если  𝑓(𝑥) = {  𝑥 + 3  , если 𝑥 |
| 2 |  | 14 | 𝑠𝑖𝑛𝑥, если 𝑥  𝑥  𝑓(𝑥) = , если  , если 𝑥 |
| 3 |  | 15 | , если 𝑥  , если    𝑠𝑖𝑛𝑥  , если 𝑥 |
| 4 |  | 16 | 𝑥|𝑥|, если 𝑥    𝑥, если    { 𝑥 , если 𝑥 |
| 5 |  | 17 |  |
| 6 |  | 18 |  |
| 7 |  | 19 |  |
| 8 |  | 20 |  |
| 9 |  | 21 |  |
| 10 |  | 22 |  |
| 11 |  | 23 |  |
| 12 |  | 24 |  |

**Задание 2.** Создайте приложение, позволяющее загружать изображения на форму.

* Создайте новый проект.
* В строке заголовка формы введите название окна программы: *Загрузка изображений*.
* На **Панели элементов** в разделе **Стандартные элементы управления** выберите компонент **Button** и разместите на форме две кнопки, в свойстве **Text** для этих кнопок запишите: «**Открыть**», «**Очистить**». Настройте свойство **Font** для этих кнопок, увеличив размер шрифта **14** пт и сделав шрифт полужирным.
* C **Панели элементов** в разделе **Стандартные элементы управления** поместите на форму компонент **PictureBox** (изображение) и компонент **OpenFileDialog** (окно диалога «Открыть файл») в разделе **Диалоговые окна** панели элементов (рисунок 54)**.**

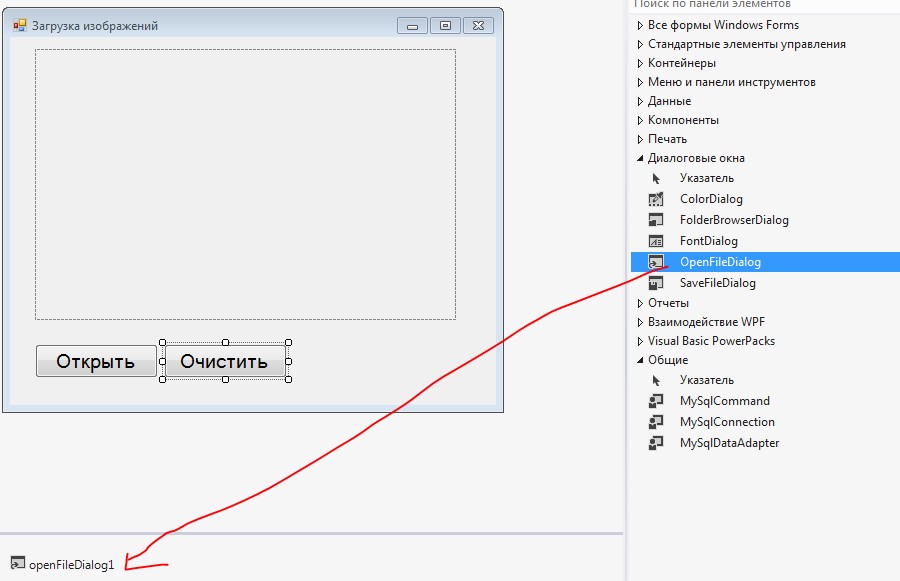


Рисунок 54- Размещение компонентов

* Настройте свойство **SizeMode** для элемента **PictureBox1** в значение **Zoom**, чтобы загружаемый рисунок масштабировался по размеру рамки.
* Создайте событие нажатие кнопки **Открыть** и между фигурных скобок введите операторы, изображенные на рисунке 55.

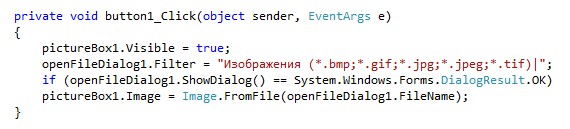


Рисунок 55 - Текст процедуры нажатия на кнопку Открыть

Первая строка отображает компонент на форме, т. к. по нажатию на кнопку очистить этот компонент будет скрываться.

Вторая строка настраивает фильтр, для отображения в окне загрузки файлов, только изображений.

Следующие строки означают: Если в окне загрузки файла нажата кнопка **ОК**, тогда загрузить в элемент изображение.

* Создайте событие для кнопки **Очистить**, запишите в теле процедуры следующий оператор (рисунок 56).

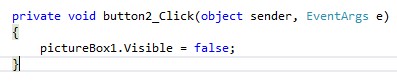


Рисунок 56- Текст процедуры нажатия на кнопку Очистить

* Запустите программу и проверьте, как она работает (рисунок 57).

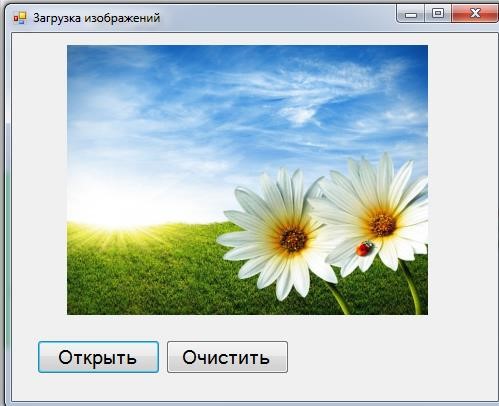


Рисунок 57-Результат тестирование проекта

**Задание 3.** Дано целое число. Если оно является положительным (сюда же относится и нулевое значение), то увеличить его на 10; в противном случае вычесть из него 5. Вывести полученное число.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Разместите компоненты, в соответствии с рисунком 58, настроив свойства данных элементов.



Рисунок 58 – Интерфейс проекта

* Создайте событие нажатие кнопки **Вычислить** (рисунок 59).

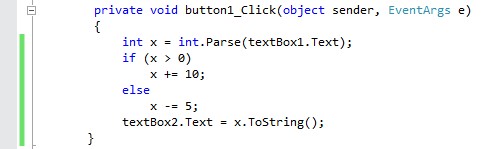


Рисунок 59 - Текст процедуры нажатия на кнопку Вычислить

* Запустите приложение на выполнение и протестируйте программу с положительными и отрицательными числами (рисунок 60).

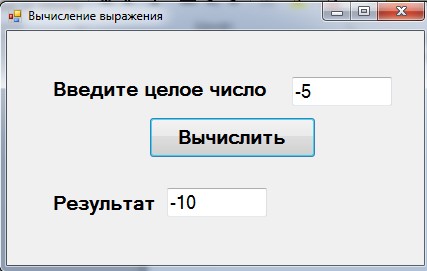
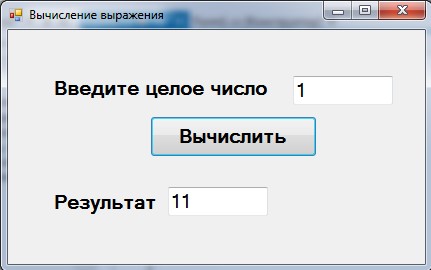


Рисунок 60 – Результат тестирования проекта

* Измените событие нажатия кнопки **Вычислить,** которое демонстрируетиспользование тернарного оператора if (рисунок 61).

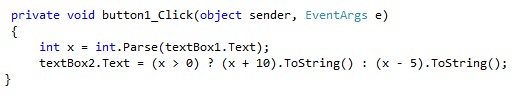


Рисунок 61 – Модифицированный текст процедуры **Самостоятельное задание № 2.** Составьте программу для решения следующих задач:

**Вариант 1.** Дано двухзначное число. Верно ли, что в его записи присутствует цифра 8?

**Вариант 2.** Дано двухзначное число. Присутствуют ли в записи этого числа цифры 2 или 7?

**Вариант 3.** Дано двухзначное число. Имеются ли в его записи цифры 3, 6 или 9?

**Вариант 4.** Дано трехзначное число. Является ли оно палиндромом (слева и справа читается одинаково)?

**Вариант 5.** Дано трехзначное число. Верно ли, что все его цифры одинаковы?

**Вариант 6.** Дано четырехзначное число. Равна ли сумма двух первых его цифр сумме двух его последних?

**Вариант 7.** Дано трехзначное число. Кратна ли сумма его цифр пяти?

**Вариант 8.** Дано трехзначное число. Кратна ли произведение его цифр числу А?

**Вариант 9.** Дано трехзначное число. Верно ли, что первая и последняя его цифры одинаковы?

**Вариант 10.** Дано трехзначное число. Верно ли, что последняя цифра этого числа является четной?

**Вариант 11.** Дано трехзначное число. Верно ли, что последняя цифра этого числа является нечетной?

**Вариант 12.** Дано четырехзначное число. Верно ли, что в записи этого числа использованы два одинаковых двузначных числа?

**Вариант 13.** Является ли первое число двухзначным, а второе однозначным?

**Вариант 14.** Является ли первое число отрицательным, а второе положительным?

**Вариант 15.** Известны два расстояния: одно в километрах, другое – в футах (1 фут=0,45 м.). Какое из расстояний меньше?

**Вариант 16.** Даны три числа. Найти среднее из них (то есть число, расположенное между наименьшим и наибольшим).

**Вариант 17.** Даны три числа. Найти сумму двух наибольших из них.

**Вариант 18.**  Дано целое число. Вывести его строку-описание вида «отрицательное четное число», «нулевое число», «положительное нечетное число» и т. д.

**Вариант 19.** Дано целое число, лежащее в диапазоне 1–999. Вывести его строку - описание вида «четное двузначное число», «нечетное трехзначное число» и т. д.

**Самостоятельное задание №3.** Составьте программу для решения следующих задач:

**Вариант 1.** Даны натуральные числа х и у, определяющие возраст двух друзей. Из данных чисел напечатать те, которые обозначают «школьный» возраст, то есть от 7 до 16 лет.

**Вариант 2.** Три поросенка: Ниф-Ниф, Нуф-Нуф и Наф-Наф — нашли под дубом желуди. Каждый из них набрал себе приличный запас на зиму (дано три числа). Составить программу, которая бы определяла, кто из поросят поленился и набрал меньше всех.

**Вариант 3.** Федя пытается протолкнуть в форточку размерами X см и У см свою голову диаметром не более D см. Удастся ли мальчику его эксперимент, или его голова застрянет?

**Вариант 4.** Даны два конверта прямоугольной формы с длинами сторон (а, b) и (с, d). Определить, можно ли один из конвертов вложить в другой?

**Вариант 5.** Имеется прямоугольное отверстие со сторонами а и b и кирпич с ребрами х, у, z. Требуется определить, пройдет ли кирпич в отверстие?

**Вариант 6.** Сестры подсчитали свои денежные накопления: у них оказались разные суммы. Какая сестра должна дать другой денег и сколько, чтобы у обеих сестер суммы стали равными?

**Вариант 7.** Напишите программу-модель анализа пожарного датчика в помещении, которая выводит сообщение «Пожарная ситуация», если температура (ее значение вводится с клавиатуры) в комнате превысила 60ºС.

**Вариант 8.** Рис расфасован в два пакета. Вес первого - m кг, второго – n кг. Составьте программу, определяющую: какой пакет тяжелее – первый или второй? вес более тяжелого пакета.

**Вариант 9.** В туристической фирме клиенту на выбор были предложены три путевки (про каждую известны стоимость N1, N2 и N3 и продолжительность тура Dl, D2 и D3 дней). Какую из трех путевок может выбрать турист, если в его распоряжении X денег и У дней отпуска?

**Вариант 10.** Жители Норвегии, Швеции и Финляндии спорили, где зима холоднее, называя среднюю температуру января в своих столицах. Определите, в каких странах эта температура оказалась ниже средней температуры января в Челябинске (t = - 13°С).

**Вариант 11.** За контрольную работу в классе учениками было получено Р пятерок, СН четверок, TR троек и DV двоек. Определить сколько человек учится в классе.

**Вариант 12.** За контрольную работу в классе учениками было получено Р пятерок, СН четверок, TR троек и DV двоек. Определить каков средний балл.

**Вариант 13.** За контрольную работу в классе учениками было получено Р пятерок, СН четверок, TR троек и DV двоек. Определить сколько учеников получили оценку, превышающую средний балл?

**Вариант 14.** Создайте проект «Угадай число!». При каждом запуске этого проекта случайным образом задумывается число от 1 до 10. Задача пользователя отгадать это число. При этом, в проекте должны даваться пояснения: «Недобор» или «Перебор».

**Вариант 15.** Год является високосным, если его номер кратен 4. однако, из кратных 100, високосными являются лишь кратные 400 (например, 1700, 1800 и 1900 – невисокосные, 2000 – високосный). Дано натуральное число N. Определить, является ли високосным год с таким номером.

**Критерии оценки:**

**«отлично»**

* правильно решены все задания с 1 по 3;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены варианты самостоятельных заданий 1-3 верно, самостоятельно.  **«хорошо»**
* задания с 1 по 3 решены с ошибками;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены верно, любые два самостоятельных заданий.

**«удовлетворительно»**

* решены задания с 1 по 3 с ошибками;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* решено верно, хотя бы одно самостоятельное задание.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы

**Практическая работа № 6 «Создание проектов с использованием оператора выбора switch и перечислений»** 2 часа

**Учебная цель:**

* 1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
  2. Научиться использовать оператор switch при решении задач.
  3. Освоить использование компонента comboBox.
  4. Научиться создавать перечисления.
  5. Получить навыки использования перечислений при написании пользовательских программ.

**Учебные задачи:**

* 1. Научиться записывать условия различной степени сложности;
  2. Создавать проекты разветвляющейся структуры на С# с использованием компонента comboBox.
  3. Создавать проекты для работы с перечислениями.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

В случае, когда нужно выполнить несколько сравнений подряд, очень удобно использовать оператор **switch … case**. Формат оператора: **switch** (выражение) { **case** Значение1:

последовательность операторов

**break**; **case** Значение2: последовательность операторов

**break**; **case** ЗначениеN:

последовательность операторов

**break**; default последовательность операторов }

Сначала вычисляется выражение в скобках за ключевым словом **switch**. Затем просматривается список меток (**case** Значение1 и т. д.) до тех пор, пока не находится метка, соответствующая значению вычисленного выражения. Далее происходит выполнение соответствующая последовательность операторов, следующих за двоеточием. Если же значение выражения не соответствует ни одной из меток оператора **switch**, то выполняется последовательность операторов, следующая за ключевым словом **default**. Допускается конструкция без слова **default**.

Когда после последовательности операторов встречается ключевое слово **break**, то выполнение оператора **break** приводит к выходу из оператора **switch** и переходу к следующему оператору программы. Если же оператор **break** отсутствует, то выполнение продолжается до первого оператора **break** ли до конца оператора **switch.**

В операторе **switch … case** в качестве выражения можно использовать либо переменную, либо выражение только определенных типов, а именно:

* любого целочисленного типа;
* символьного типа (**char**);  строкового типа (**string**);  перечисления. В операторе **switch … case** нельзя использовать переменные вещественного типа или типа **decimal**.

При работе с оператором **switch … case** удобно использовать такие компоненты как: **comboBox** и **listBox**. Основные свойства этих компонентов приведены в таблице 7.

Таблица 7 – Свойства компонентов comboBox и listBox

|  |  |
| --- | --- |
| Свойство | Описание |
| Items | Элементы списка - коллекция строк |
| Items.Add(«элемент») | Добавление элемента в список. |
| SelectedIndex | Номер выбранного элемента в списке. Если ни один элемент не выбран, значение свойства **=-1**. |
| SelectedItem | Выбранный элемент списка.  Если необходимо воспользоваться его значением, то переведите его в строковый тип **SelectedItem.ToString()** |
| Text | Строка текста, отображаемая в заголовке |

Главное событие, обрабатываемое при выборе элемента в любом из этих списков– **SelectedIndexChanged**.

Перечисление - это множество именованных целочисленных констант.

Ключевое слово **enum** объявляет перечислимый тип. Формат записи перечисления таков: enum имя {список\_перечисления};

где имя - указывает на имя типа перечисления, список\_перечисления – это

список идентификаторов, разделенных запятыми. Пример: enum Color {Red, Green, Blue}

Каждый элемент перечисления имеет связанное с ним константное значение, тип которого определяется базовым типом перечисления.

**Вопросы для закрепления теоретического материала:**

1. Опишите принцип работы оператора switch.
2. Сформулируйте недостатки и преимущества использования оператора switch.
3. Назовите свойство компонента comboBox, позволяющее узнать индекс выбранного элемента.

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Составить программу, выводящую картинку по выбранному времени года.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Работа** **с ComboBox**.
* Разместите компонент **ComboBox**. В свойстве **Font** установите полужирное начертание и размер шрифта равным **14** пт.
* В свойстве **Items** задайте значение списка: **Зима, Весна, Лето, Осень** (рисунок 62).

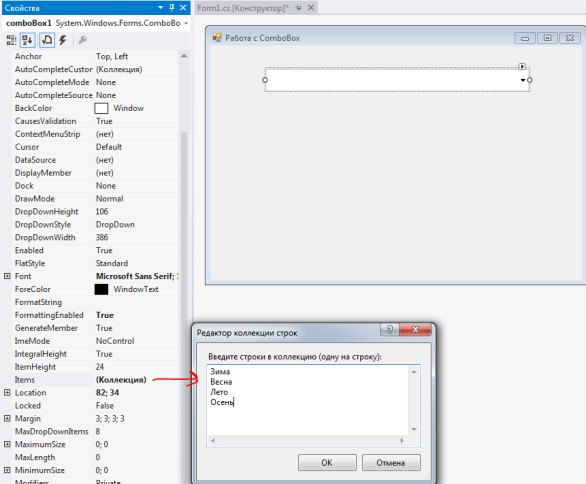


Рисунок 62 – Строки элемента ComboBox1

* В свойсте **Text** задайте значение **Выберите время года** (Рисунок 63).

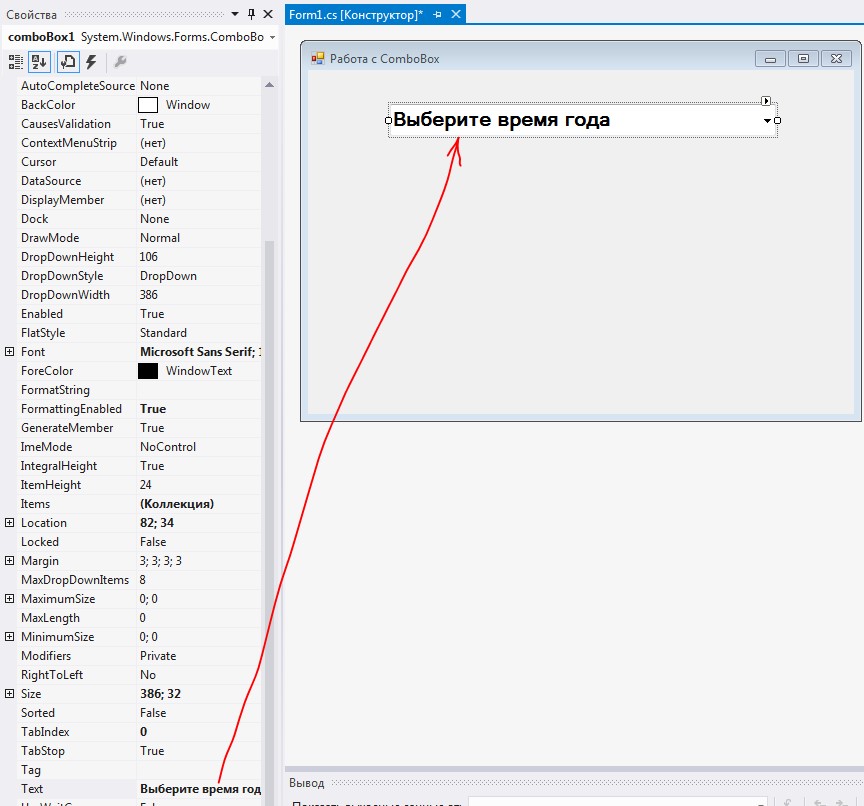


Рисунок 63 – Свойство Text элемента ComboBox1

* Разместите на форме элемент **PictureBox**. Установите свойство **SizeMode** в значение **Zoom** (Рисунок 64).

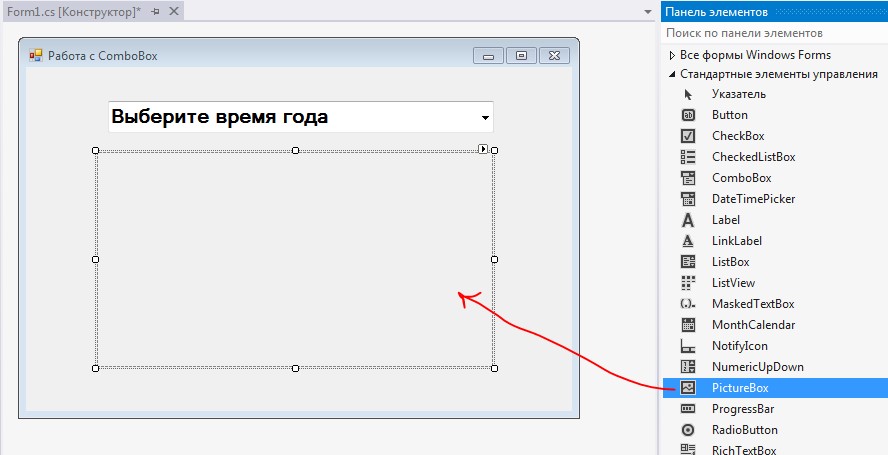


Рисунок 64 – Размещение элемента PictureBox

* В папку с проектом из Интернета скачайте изображения разного вида времен года и переименуйте в соответствии времени года (Рисунок 65).

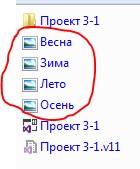


Рисунок 65 – Папка проекта

* Создайте событие выбора элемента списка для элемента **comboBox1**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 66).

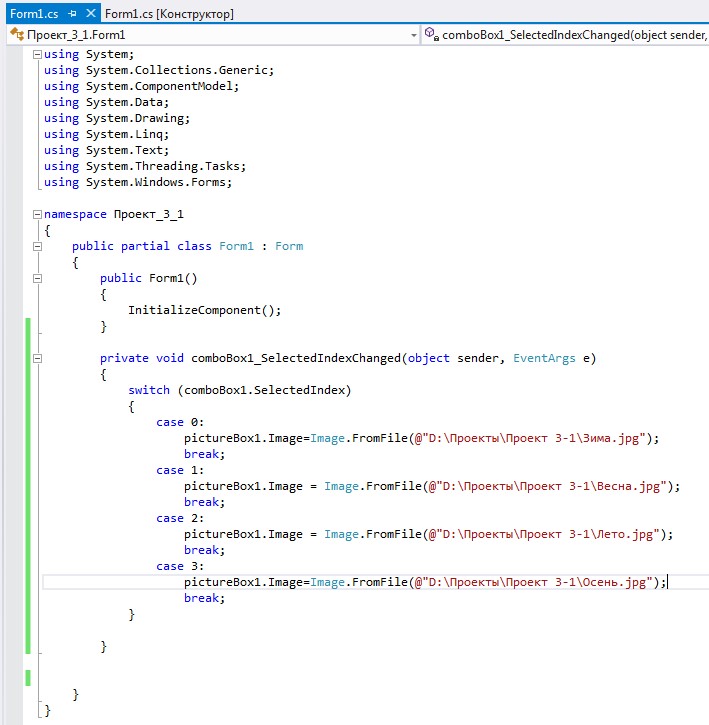


Рисунок 66 - Текст процедуры выбора элемента списка

* Запустите приложение на выполнение и протестируйте программу (Рисунок 67).

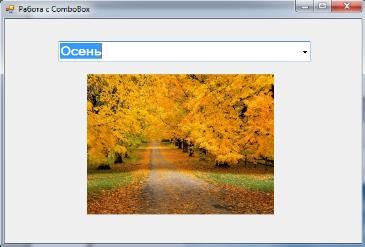
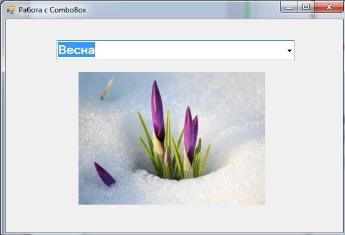
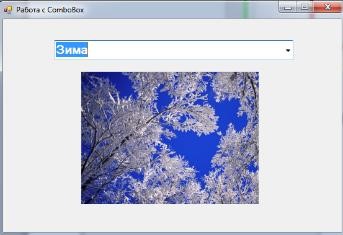
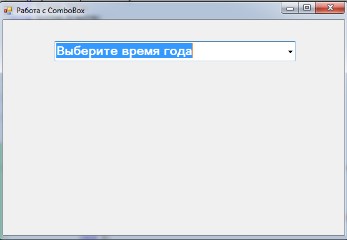


Рисунок 67 – Результат тестирования проекта

**Самостоятельное задание №1**. Создайте новый проект для решения задач по варианту:

**Вариант 1.** Дано целое число K. Вывести строку-описание оценки, соответствующей числу K (1 — «плохо», 2 — «неудовлетворительно», 3 — «удовлетворительно», 4 — «хорошо», 5 — «отлично»). Если K не лежит в диапазоне 1– 5, то вывести строку «ошибка».

**Вариант 2.** Дан перечень месяцев года. Вывести название соответствующего времени года («зима», «весна», «лето», «осень»), сопроводив все соответствующее картинками.

**Вариант 3.** Дан перечень месяцев года. Определить количество дней в выбранном месяце. В проекте следует вводить год и выдавать количество дней в месяце в зависимости от года.

**Вариант 4.** Дан перечень арифметических операций: «+» — сложение, «-» — вычитание, «\*» — умножение, «/» — деление. Также даны вещественные числа A и B (B не равно 0). Выполнить над числами указанное действие и вывести результат.

**Вариант 5.** Дан перечень единиц длины (миллиметр, сантиметр, метр, дециметр, километр) и длина отрезка в этих единицах (вещественное число). Найти длину отрезка в метрах.

**Вариант 6.** Дан перечень единиц массы (миллиграмм, грамм, килограмм, центнер, тонна) и масса тела в этих единицах (вещественное число). Найти массу тела в килограммах.

**Вариант 7.** Известен радиус окружности и имеется перечень элементов, касающихся этой окружности (диаметр D = 2·R, длина L = 2·π·R, площадь круга S = π·R2). Определить значение для выбранного элемента данной окружности.

**Вариант 8.** Известны длины сторон прямоугольника. Кроме того, имеется перечень элементов, касающихся этого прямоугольника: периметр, площадь, диагональ. Необходимо определить значение для выбранного элемента прямоугольника.

**Вариант 9.** Дано целое число в диапазоне от1 до 99, определяющее возраст человека (в годах). Вывести фразу «мне N лет», обеспечив правильное согласование числа со словом «год», например: 20 — «двадцать лет», 32 — «тридцать два года», 41 — «сорок один год».

**Вариант 10.** Написать программу, позволяющую по последней цифре числа определить последнюю цифру его квадрата.

**Вариант 11.** Дано натуральное число n (1 ≤ n ≤ 9999), определяющее стоимость товара в копейках. Выразить стоимость в рублях и копейках. Например, 3 рубля 21 копейка, 15 рублей 5 копеек, 1 рубль ровно и т.п.

**Вариант 12.** В старояпонском календаре был принят 60-летний цикл, состоявший из пяти 12-летних подциклов. Подциклы обозначались названиями цвета: зеленый, красный, желтый, белый и черный. Внутри каждого подцикла годы носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи (для справки: 1984 – год зеленой крысы – был началом очередного цикла). Напишите программу, которая вводит номер некоторого года нашей эры и печатает его название по старояпонскому календарю.

**Вариант 13.** На форме имеется 3 кнопки: на одной написано – «воздушный транспорт», на другой – «наземный транспорт», на третьей – «водный транспорт». Нажатие каждой из этих кнопок должно сопровождаться отображением соответствующей иллюстрации.

**Вариант 14.** На форме имеется 6 кнопок, на каждой из которых написаны числа от 1 до 6 (позиции игрального кубика). Нажатие каждой из этих кнопок должно сопровождаться отображением соответствующей иллюстрации.

**Вариант 15.** Для каждой введённой цифры (0-9) вывести соответствующее ей название на английском языке (0-zero, 1-one, 2-two, …).

**Вариант 16.** Составить программу, позволяющую получит словесное описание школьных отметок (1-«плохо», 2-«неудовлетворительно», 3-«удовлетворительно», 4-«хорошо», 5-«отлично»).

**Вариант 17.** Напишите программу, позволяющую рассчитать стоимость жалюзи в зависимости от материала изготовления и размеров этих жалюзи. Для этого разместите на форме комбинированный список и заполните его названиями материала изготовления (пластик, алюминий, соломка, текстиль). Разместите также элементы, позволяющие вводить размеры жалюзи в сантиметрах. Выбор элемента списка должен приводить к демонстрации цены за 1 кв.м. и расчетам по введенным размерам.

**Вариант 18.** Напишите программу, позволяющую рассчитать стоимость фотографии в зависимости от ее размеров. Для этого разместите на форме комбинированный список и область для размещения изображений. Список нужно будет заполнить возможными размерами фотографии. При выборе элемента списка: рассчитывается стоимость этой фотографии (известна цена за 1 кв. см.) и меняется размер изображения.

**Задание 2.** Создать проект, имитирующий выбор сигнала светофора.

Составить программу, выводящую картинку по выбранному времени года.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Светофор**.
* Разместите компонент **ComboBox**. В свойстве **Font** установите полужирное начертание и размер шрифта равным **12** пт.
* В свойстве **Items** задайте значение списка: **Красный, Желтый, Зеленый**  (Рисунок 68).

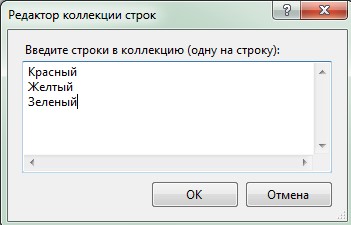


Рисунок 68- Значения свойства Items

* Разместите компонент **Label**. Удалите значения из свойства **Text** и в свойстве **Font** установите полужирное начертание и размер шрифта равным **12** пт (Рисунок 69).

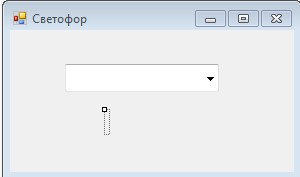
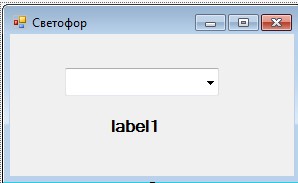


Рисунок 69 – Размещение компонентов на форме

* Создайте событие выбора элемента списка для элемента **comboBox1**, для этого достаточно дважды щелкнуть мышью по данному элементу. Не забудьте объявить тип перечисление перед методом нажатия кнопки (Рисунок 70).



Рисунок 70 – Код проекта

* Запустите приложение на выполнение и протестируйте программу (Рисунок 71).

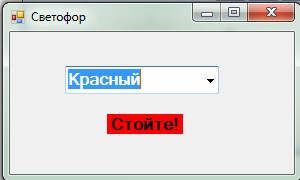


Рисунок 71 – Резальтат тестирования приложения

**Самостоятельное задание №2**. Создайте новый проект для решения задач по варианту:

**Вариант 1.** Создайте проект, который позволял бы менять оформление визуальных элементов на форме (использовать перечисления)

**Вариант 2.** Создайте проект, позволяющий выводить название дня недели по его номеру, используя перечисление.

**Вариант 3.** Создайте проект, позволяющий по дате рождения определить знак зодиака и увидеть картинку этого знака (использовать перечисление).

**Критерии оценки:**

**«отлично»**

* правильно решено задание 1 и 2;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены варианты самостоятельных заданий 1-2 верно, самостоятельно.

**«хорошо»**

* правильно решено задание 1 и 2;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решен один из вариантов самостоятельных заданий верно и номер варианта совпадает с заданным.

**«удовлетворительно»**

* правильно решено задание 1 и 2;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.  не решен ни один вариант самостоятельных заданий.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы.

**Практическая работа № 7 «Создание проектов создание циклической структуры (for)»**

4 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить построение циклических алгоритмов с известным количеством проходов.
3. Освоить использование циклического оператора for при решении задач.
4. Освоить использование компонента listBox. **Учебные задачи:**
5. Научиться создавать проекты различной сложности с использованием циклов.
6. Создавать проекты разветвляющейся структуры на С# с использованием компонента listBox.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Циклический оператор **for** используется в том случае, когда количество повторений либо заранее известно, либо может быть четко определено.

Синтаксис оператора: **for** (инициализация; проверка условия; изменение) оператор;

Пример использования:

int s = 0;

for (int i = 1; i<=100; i++) s+=i;

В простейшей форме инициализация используется для присвоения начального значения параметру цикла. Проверка условия - обычно условное выражение, которое определяет, когда цикл должен быть завершен. Приращение обычно используется для изменения параметра цикла каждый раз при повторении цикла. Эти три раздела заголовка цикла должны быть разделены точкой с запятой. Выполнение цикла происходит до тех пор, пока условное выражение истинно. Как только условие становится ложным, начинает выполняться следующий за циклом **for** оператор.

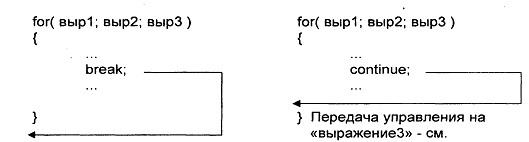


Рисунок 72 – Схема выполнения оператора Схема выполнения оператора **for** (рисунок 72):

1. Вычисляется выражение 1.
2. Вычисляется выражение 2.
3. Если значения выражения 2 отлично от нуля (истина), выполняется тело цикла, вычисляется выражение 3 и осуществляется переход к пункту 2, если выражение 2 равно нулю (ложь), то управление передается на оператор, следующий за оператором **for**.

Проверка условия всегда выполняется в начале цикла. Это значит, что тело цикла может ни разу не выполниться, если условие выполнения сразу будет ложным.

Любое из трех выражений может быть опущено, но точки с запятой обязательно должны стоять. Если пропущено выражение2, считается, что оно имеет значение 1 /истинно/, и цикл никогда не заканчивается /бесконечный цикл/.

Оператор **break** — завершает выполнение цикла, внутри которого записан.

Оператор **continue** — выполняет переход к следующей итерации цикла.

Важной функцией класса **String**, является функция **Format**. Эта функция формирует строку из нескольких составляющих и с определенным форматированием. Синтаксис функции **Format** следующий:

String.Format(строка содержащая шаблон, param[] аргумент) param[] говорит методу, что он может принимать неограниченное

количество аргументов, разделенных запятой. На примере это выглядит так:

String.Format(“{0} {1}”, “Привет”, “всем”);

Вернет строку «Привет всем». В фигурных скобках указываются индексы полей, которые передаются, начиная с нуля. «Привет» — нулевое поле, «всем» — первое поле. В самих фигурных скобках, кроме форматирования через запятую можно указывать тип или, другими словами, как форматировать данное поле.

При выводе чисел индексам можно задавать типы форматирования в виде букв (через двоеточие) (таблица 8).

Таблица 8 – Форматирование строк

|  |  |
| --- | --- |
| **Функция** | **Результат** |
| String.Format("{0:c}",12.53) | 12,53р. |
| String.Format("{0:e}",12.53) | 1,253000e+001 |
| String.Format("{0:f}", 12.53) | 12,53 |
| String.Format("{0:00.0000}", 12.53) | 12,5300 |
| String.Format("{0:(#).##}", 12.53) | (12),53 |
| String.Format("{0:0,0}", 12.53) | 13 |
| String.Format("{0:000.0}", 12.53) | 012,5 |
| String.Format("{0:0,0%}", 12.53) | 1 253% |
| String.Format("{0:$#,##0.00;($#,##0.00);Zero}",  12.53) | $12,53 |
| String.Format("{0:$#,##0.00;($#,##0.00);Zero}", 1245.53) | $1 245,53 |
| String.Format("{0:d}", DateTime.Now) | 13.07.2015 |
| String.Format("{0:t}", DateTime.Now) | 13:00 |
| String.Format("{0:g}", DateTime.Now) | 13.07.2015 13:00 |

**Вопросы для закрепления теоретического материала:**

1. В каких случаях целесообразно использовать цикл **FOR**?
2. Пусть цикл записан в виде: **FOR** (**int** v:=e1; v<=e2; v++) S; Как называется e1?

**Задания и инструкция по выполнению практической работы Задание 1.** Вывести таблицу значений функции y=sin(x), если х меняется от 1 до 2 с шагом 0.2.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Таблица значений F(x)**.
* Разместите компонент **ListBox**. В свойстве **Font** установите полужирное начертание и размер шрифта равным **14** пт (Рисунок 73).

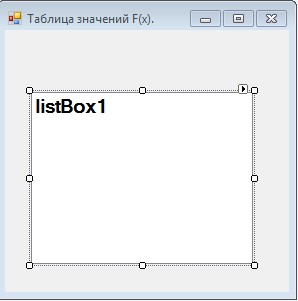


Рисунок 73 – Интерфейс приложения

* Разместите на форме элемент **Button**. В свойстве **Font** установите полужирное начертание и размер шрифта равным **14** пт. В свойстве **Text** установите значение **Пуск** (Рисунок 74).

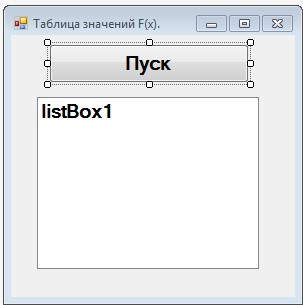


Рисунок 74 – Интерфейс приложения

* Создайте событие для элемента **Button1**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 75).

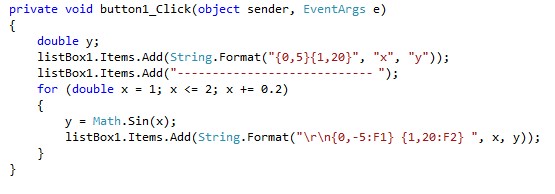


Рисунок 75 – Код проекта

* Запустите приложение на выполнение и протестируйте программу (Рисунок 76).



Рисунок 76-Результат тестирования приложения

* Экспериментальным путем выясните значение каждого параметра функции Format (удалите один симво, запусмтиет и сравните результаты, верните символ, затем удалите другой и т. д.). Результат исследования запишитие в тетрадь.

**Задание 2.** Известно N чисел (числа задаются случайным образом в промежутке [-50; 50]). Составьте программу, которая определяет количество отрицательных, количество положительных и количество нулей среди введенных чисел. Значение N вводится с клавиатуры.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Определение** **чисел**.
* Разместите компоненты **ListBox, Button, TextBox, Label** как показано на рисунке. В свойстве Font установите размер шрифта равным 14 пт (Рисунок 77).

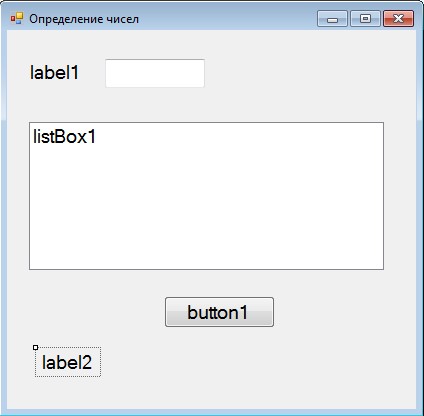


Рисунок 77 – Размещение элементов на форме проекта - В свойствах Text задайте значения, соответствующие рисунку 78.

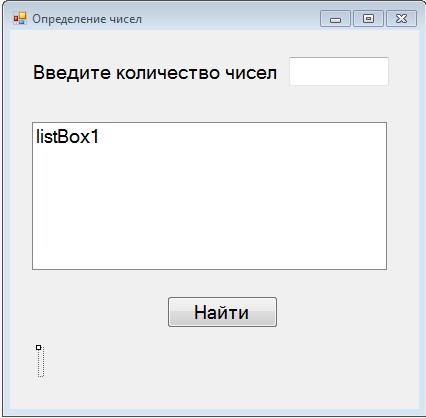


Рисунок 78 – Интерфейс проекта

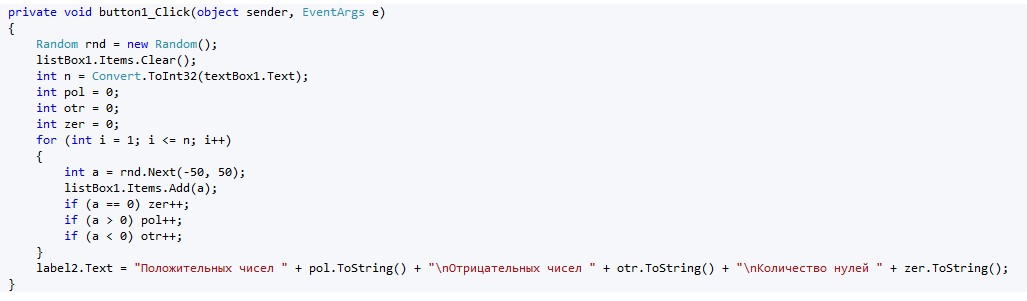
* Создайте событие для элемента Button1, для этого достаточно дважды щелкнуть мышью по данному элементу.

Алгоритм данного события:

1. Создается новый экземпляр класса Random (генератора случайных чисел).
2. Очищается ListBox1.
3. Объявление переменных: n – целый тип и первоначальное значение берет из компонента TextBox1, предварительно преобразовав тип; pol – количество положительных чисел; otr - количество отрицательных чисел; zer – количество нулей.
4. В цикле от 1 до n выполняется:

* Присваивание случайного числа целочисленной переменной a;
* Добавление элемента в компонент ListBox1;
* Если а равна нулю, тогда переменная zer увеличивается на единицу;
* Если а положительная, тогда переменная pol увеличивается на единицу;
* Если а отрицательная, тогда переменная otr увеличивается на единицу;

5. Выводиться результат в элемент Label 2. \n означает вывод с новой строки (Рисунок 79).



2

3

4

5

1

Рисунок 79 – Код проекта

* Запустите приложение на выполнение и протестируйте программу.

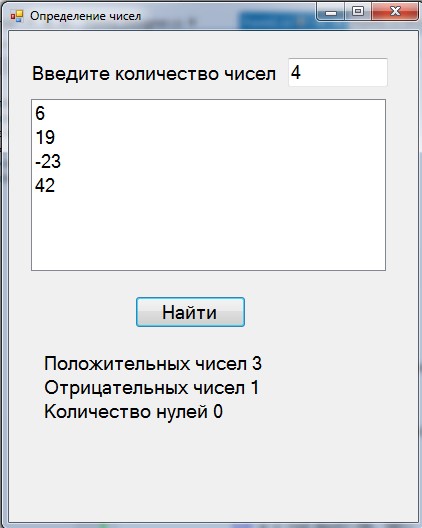


Рисунок 80- Результат тестирования приложения

**Задание 3.** Ввести с клавиатуры 10 чисел. Если среди них есть числа, большие 15, заменить их на 15.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите ОК.
* Измените заголовок формы, в свойстве Text установите значение Определение чисел.
* Разместите компоненты ListBox, Button, TextBox, Label как показано на рисунке. В свойстве Font установите размер шрифта равным 12 пт.
* В свойствах Text задайте значения, соответствующие рисунку 81.

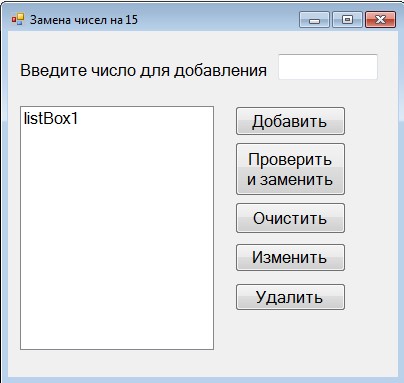
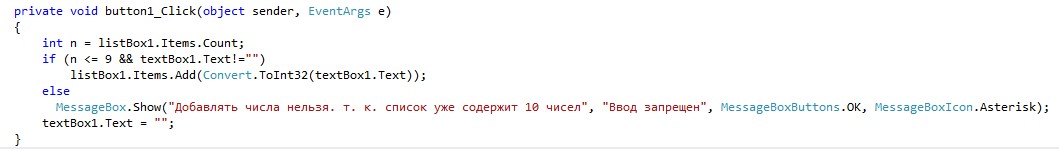


Рисунок 81 – Интерфейс проекта

* Создайте событие для элемента Button1 (кнопка Добавить), для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 82).

Алгоритм данного события:

1. Объявляется целочисленная переменная n, начальное значение которой равно количеству строк элемента ListBox1.
2. Если n (количество элементов списка) меньше или равно 9 (т. к. номера строк в списке ListBox начинают нумероваться с нуля) и если текстовый блок пустой, то добавить строку в список ListBox1 из элемента TextBox1 преобразовав значение к целому типу, иначе вывести сообщение о невозможности ввода чисел в список.



2

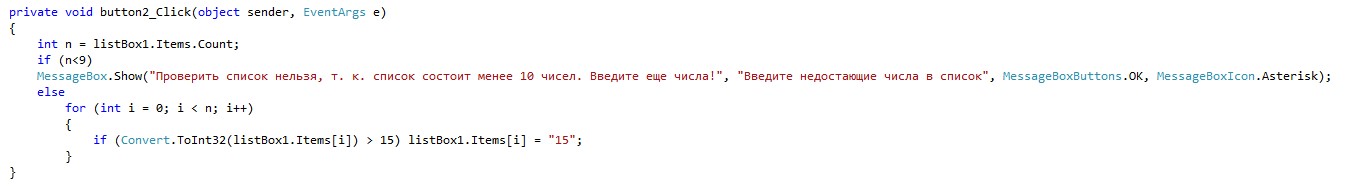
1

Рисунок 82 – Код нажатия на кнокпку Добавить

- Создайте событие для элемента Button2 (кнопка Проверить и изменить), для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок

83). Алгоритм данного события:

1. Объявляется целочисленная переменная n, начальное значение которой равно количеству строк элемента ListBox1.
2. Если n (количество элементов списка) меньше 9(т. к. номера строк в списке ListBox начинают нумероваться с нуля) тогда вывести сообщение о недостаточном количестве строк в списке, иначе в цикле от i=0 до n проверять если значение строки с номером i больше 15, тогда изменить данную строку на значение равное 15.



1

2

Рисунок 83 – Код нажатия на кнопку Проверить и изменить

* Создайте событие для элемента Button3 (кнопка Очистить), для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 84).



Рисунок 84 – Код нажатия на кнопку Очистить

* Создайте событие для элемента Button4 (кнопка Изменить), для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 85).

Данное событие содержит один условный оператор: Если элемент TextBox1 пустой, тогда вывести сообщение о необходимости ввести значение на которое следует поменять строку списка, иначе выделенному номеру строки списка присвоить значение из элемента TextBox1, преобразовав к типу Int.

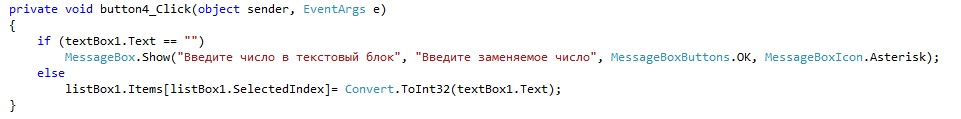


Рисунок 85 – Код нажатия на кнопку Изменить

* Создайте событие для элемента Button4 (кнопка Удалить), для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 86). RemoveAt – удаляет элемент по указанному индексу.

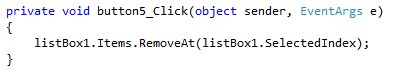


Рисунок 86 – Код нажатия на кнопку Удалить

**Самостоятельное задание №1.** Создайте новый проект для решения одной из задачи по вариантам.

**Вариант 1.** Белоснежка на Рождество решила купить гномам новые рубашки и измерила рост каждого. В магазине продаются рубашки трех ростов: I — 2535 см, II — 3645 см и III — 4655 см. Выведите таблицу, в которой будут указаны реальный рост каждого гнома и соответствующий ему рост рубашки.

**Вариант 2.** При поступлении в ВУЗ абитуриенты, получившие двойку на первом экзамене, ко второму не допускаются. Известны оценки десяти абитуриентов, полученные на первом экзамене. Определить, кто из них допущен ко второму экзамену.

**Вариант 3.** В чемпионате по футболу команде за выигрыш дается 3 очка, за ничью – 2 очка, за проигрыш – 0 очков. Известно количество очков, полученных каждой из N команд. Определите словесный итог игры для каждой команды.

**Вариант 4.** Петру Петровичу очень везло на рыбалке. Начиная с понедельника, он каждый день ловил К золотых рыбок (случайное число от 0 до 4) и каждая исполняла по 3 его желания. Сколько желаний Петра Петровича сбывалось ежедневно в течение всей недели.

**Вариант 5.** Известно количество абитуриентов, подавших заявления на каждую из десяти специальностей и план приема по каждой специальности. Определить конкурс по каждой специальности.

**Вариант 6.** Известны цены двенадцати наименований товаров в магазине. Покупатель имеет N рублей и может купить только один товар. Определитесь по каждому товару, сможет ли он его приобрести.

**Вариант 7.** Из двух собачьих будок, находящихся на расстоянии 27 км одна от другой, навстречу друг другу выскочили в одно и то же время две драчливые собачки. Первая бежит со скоростью 4 км/ч, а вторая – 5 км/ч. Выдайте таблицу, из которой будет видно, как быстро сокращается расстояние между собачками: отобразите (ежечасно) расстояние, которое пробегают собачки, и расстояние, которое еще остается между ними.

**Вариант 8.** Племя каннибалов решило съесть своих соседей, но дикари не знали, какое из ближайших племен выбрать: если в племени мало народу, то на всех каннибалов не хватит; если много — самих каннибалов побьют. Оптимальное количество известно: от 10 до 65 человек. Рядом с ними проживало К племен, в каждом из которых известно количество жителей. Какие из этих племен людоеды могут съесть?

**Вариант 9.** После уроков N учеников решали и отлаживали задачи на компьютере. Первый полностью отладил задачу за К минут (число вводится), все последующие тратили времени на 10 минут больше предыдущего. Сколько компьютерного времени им понадо6илось?

**Вариант 10.** У прилавка магазина выстроилась очередь из 8 покупателей. Вам известно время обслуживания каждого покупателя. Определить время пребывания в очереди конкретного покупателя.

**Вариант 11.** Клиент внес в банк К рублей. Каждый месяц эта сумма увеличивается на Y процентов. Выдать таблицу ежемесячных выплат клиента.

**Вариант 12.** Составить программу вычисления суммы квадратов всех натуральных чисел от 1 до N.

**Вариант 13.** Составить программу вычисления произведения всех четных натуральных чисел от 1 до N.

**Вариант 14.** Найти произведение N произвольных чисел.

**Вариант 15.** Известно N чисел (числа задаются случайным образом в промежутке [-15; 10]). Определить сумму положительных чисел и произведение чисел, кратных пяти.

**Вариант 16.** Известно N чисел (числа задаются случайным образом в промежутке [-50; 50]). Определить сумму отрицательных четных чисел и произведение чисел, кратных трем.

**Вариант 17.** Известно N чисел (числа задаются случайным образом в промежутке [1; 50]). Сравнить произведение четных чисел и произведение чисел, кратных трем или четырем.

**Вариант 18.** Известно N чисел (числа задаются случайным образом в промежутке [20; 50]). Вывести те из них, которые делятся на 3, но не делятся на 5.

**Вариант 19.** Вычислить произведение целых нечетных чисел от M до N (M<N).

**Вариант 20.** Дано натуральное n. Вычислить 1\*3\*5\*....\*n - для нечетных n и 2\*4\*6\*....\*n - для четных n.

**Вариант 21.** Ввести с клавиатуры 10 чисел – положительных и отрицательных. Заменить все отрицательные числа их модулями и напечатать все полученные 10 чисел.

**Вариант 22.** Вывести двузначные числа, которые делятся на 4, но не делятся на 6.

**Вариант 23.** Дано вещественное число — цена 1 кг конфет. Вывести стоимость 1, 2, … , 10 кг конфет.

**Вариант 24.** Для температуры от -15о до +30оС найти соответствующие значения температуры по шкале Фаренгейта и Кельвина. Известно, что для пересчета по шкале Фаренгейта нужно исходное значение t°С умножить на 1,8 и прибавить 32, а по шкале Кельвина значение абсолютного нуля соответствует — 273,15оС.

**Вариант 25.** Известна цена 1 кг изюма (значение вводится с клавиатуры).

Вывести стоимость 100, 200, …, 1000 г изюма.

**Вариант 26.** Напечатать таблицу перевода расстояний в дюймах в сантиметры (1 дюйм = 2,54 см) для значений от 2 до 12 с шагом в 1 дюйм (от K до N дюймов с шагом H).

**Вариант 27.** Напечатать таблицу соответствия между весом в фунтах и весом в килограммах для значений от 1 до N фунтов с шагом H фунт (1 фунт = 400 гр.).

**Вариант 28.** Пожарных учат надевать штаны за 10 секунд. Получите таблицу одевания одного пожарного в течение пяти минут.

**Самостоятельное задание №2.** Составить программу выводящую список по варианту.

**Вариант 1.** всех четных чисел от 1 до 100.

**Вариант 2.** всех нечетных чисел от 1 до 100.

**Вариант 3.** всех чисел, заканчивающихся на цифру 5, от 1 до 100.

**Вариант 4.** всех чисел, заканчивающихся на цифру 2, от 1 до 100.

**Вариант 5.** всех чисел, заканчивающихся на цифру 3, от 1 до 100.

**Вариант 6.** всех чисел, заканчивающихся на цифру 4, от 1 до 100.

**Вариант 7.** всех четных чисел от 1 до 1000.

**Вариант 8.** всех нечетных чисел от 1 до 1000.

**Вариант 9.** всех чисел, заканчивающихся на цифру 5, от 1 до 1000.

**Вариант 10.** всех чисел, заканчивающихся на цифру 2, от 1 до 1000.

**Вариант 11.** всех чисел, заканчивающихся на цифру 3, от 1 до 1000.

**Вариант 12.** всех чисел, заканчивающихся на цифру 4, от 1 до 1000.

**Вариант 13.** всех чисел, заканчивающихся на цифру 6, от 1 до 1000.

**Вариант 14.** всех чисел, заканчивающихся на цифру 7, от 1 до 1000. **Вариант 15.** всех чисел, заканчивающихся на цифру 8, от 1 до 1000.

**Критерии оценки:**

**«отлично»**

* правильно решены все задания с 1 по 3;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены варианты самостоятельного задания №1,2, самостоятельно.

**«хорошо»**

* задания с 1 по 3 решены с ошибками;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены варианты самостоятельного задания №1,2, с подсказками преподавателя.

**«удовлетворительно»**

* решены задания с 1 по 3 с ошибками;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* решено одно из самостоятельных заданий.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы

**Практическая работа № 8 «Создание проектов создание циклической структуры (while и do)»**

4 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить построение циклических алгоритмов с не известным количеством проходов.
3. Освоить использование циклических операторов while и do при решении задач.

**Учебные задачи:**

1. Научиться создавать проекты различной сложности с использованием циклов.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Циклический оператор **while** предназначен для организации цикла с предусловием, а оператор **do** – для организации цикла с постусловием. Выбор подходящего циклического оператора зависит от решаемой задачи, а также от личных предпочтений программиста.

Синтаксис цикла с предусловием: **while (выражение) оператор;**

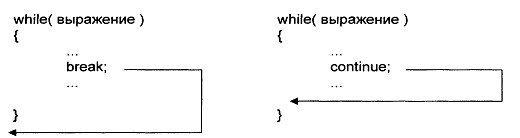


Рисунок 87 – Схема работы оператора цикла while

Схема выполнения оператора **while** следующая:

* 1. Вычисляется выражение.
  2. Если выражение ложно, то выполнение оператора **while** заканчивается и выполняется следующий по порядку оператор. Если выражение истинно, то выполняется тело оператора **while**. 3. Процесс повторяется с пункта 1.

Цикл **while** является «условным» циклом, использующим условие на входе.

При построении цикла **while** необходимо включить в него какие-то конструкции, изменяющие величину проверяемого выражения так, чтобы, в конце концов, оно стало ложным. В противном случае выполнение цикла никогда не завершится.

Для изменения хода выполнения операторов, составляющих тело цикла можно воспользоваться операторами **break** и **continue**.

Синтаксис цикла с постусловием: **do оператор while выражение;**

Оператор **do** повторно выполняет оператор или блок операторов, пока определенное выражение не примет значение **false**. В других языках программирования (Паскаль) данный чикл работает иначе.

Рекомендации по написанию циклов

* если в теле циклов **while** и **for** требуется выполнить более одного оператора, нужно заключать их в блок;
* всем переменным, встречающимся в правой части операторов присваивания в теле цикла, до этого присвоены значения, а также возможно ли выполнение других операторов;
* проверить, изменяется ли в теле цикла хотя бы одна переменная, входящая в условие продолжения цикла.

**Вопросы для закрепления теоретического материала:**

1. Для чего нужен циклический алгоритм?
2. Чем отличается цикл с постусловием от цикла с предусловием?

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Заполните таблицу 9 в тетради.

Таблица 9 – Таблица для сравнения операторов цикла

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **For** | **While** | **Do** |
| Минимальное количество раз выполнения |  |  |  |
| Операторные скобки {} |  |  |  |
| Выход из цикла false/true |  |  |  |
| Изменение переменной условия |  |  |  |
| Задание начального значения |  |  |  |
| Место нахождения тела цикла |  |  |  |

**Задание 2.** Создайте проект, определяющий количество цифр в числе.

Количество цифр в числе **n** неизвестно, поэтому необходимо использовать оператор **While**. Использование **For** потребует или введения дополнительных переменных, или искусственного выхода из цикла. Подсчет количества цифр начнем с последней цифры числа. Увеличим счетчик цифр на единицу (**k**). Число (**n**) уменьшим в 10 раз (разделив его на 10 и взяв только целую часть от деления), убирая тем самым из него последнюю цифру (подсчитанную). Далее с получившимся числом проделаем ту же последовательность действий и т. д., пока число не станет равным нулю.

Пусть введено число 65387, присвоим это значение переменной с именем m, значение счетчика числа цифр (**k**) равно 0. Выполним действия, описанные выше, их результат приведен в таблице 10. Итак, окончательное значение переменной **k** равно 5, в числе и цифр.

Таблица 10 – Тестирование алгоритма

|  |  |
| --- | --- |
| k | n |
| 0 | 65387 |
| 1 | 6538 |
| 2 | 653 |
| 3 | 65 |
| 4 | 6 |
| 5 | 0 |

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Определение характеристик числа**.
* Разместите компоненты **Button, TextBox, Label** как показано на рисунке 88. В свойстве **Font** установите размер шрифта равным **12** пт.

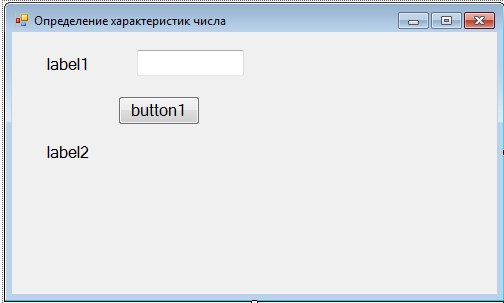


Рисунок 88 – Размещение элементов на форме проекта

* В свойствах **Text** задайте значения, соответствующие рисунку 89.

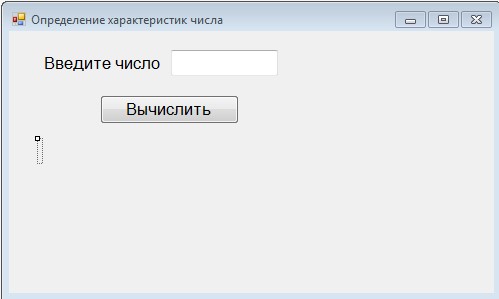
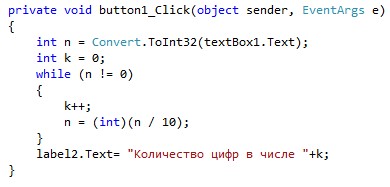


Рисунок 89- Интерфейс приложения

* Создайте событие для элемента **Button1**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 90).



объявляется целая переменная n, начальное значе-

ние которой равно значению в TextBox1, предвари-

тельно преобразовав к целому типу

Пока n (число) не

равно нулю

Количество найденных чисел увеличиваем на единицу

Откидываем последнюю цифру числа, поделив его

на 10 и взяв только целую часть

Рисунок 90 – Код проекта

* Запустите приложение на выполнение и протестируйте программу (Рисунок 91).

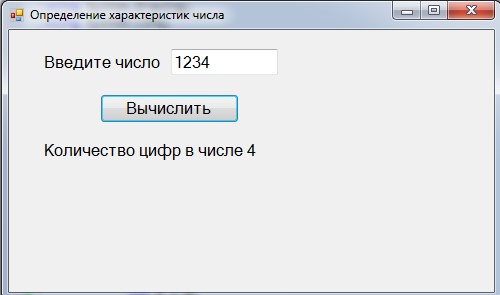


Рисунок 91 – Результат тестирования приложения

* Модифицируйте программу, так чтобы она еще находила сумму цифр числа. Добавьте на форму еще один **Label**, для вывода суммы числа. В свойстве **Font** установите размер шрифта равным **12** пт (Рисунок 92).

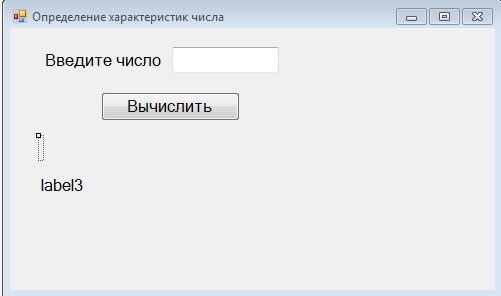


Рисунок 92-Добавление метки на форму

* Удалите у данного элемента значение свойстве **text** (Рисунок 93).

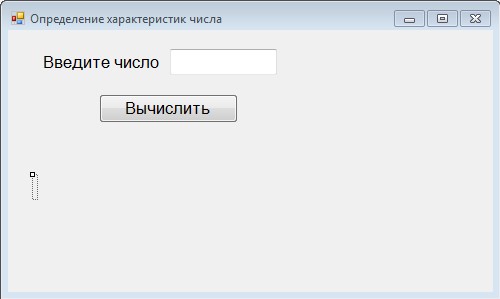


Рисунок 93-Добавлен новый элемент

* Откройте ранее созданное событие для элемента **Button1**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 94). Дополните программу вычислением суммы цифр числа. Операция **%** вычисляет остаток от деления, т. е. поделив число на 10, мы будим добавлять его к сумме.



объявляется целая переменная s для сохранения

суммы цифр числа

К предыдущему значению суммы прибавляем

последнюю цифру в числе, разделив число на

10

и взяв остаток от деления

Рисунок 94 – Модифицированный код проекта

* Запустите приложение на выполнение и протестируйте программу (Рисунок 95).

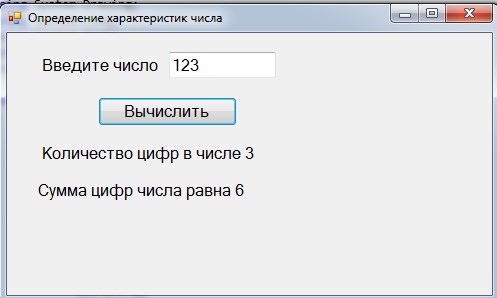


Рисунок 95 – Результат тестирования приложения

**Самостоятельное задание №1.** Продолжите модификацию программы в задании 1, так чтобы она позволяла:

­ найти первую цифру числа, например, для числа 7265 это цифра 7; ­ поменять порядок цифр числа на обратный. Например, было 12345, стало

54321;

­ найти количество четных цифр числа;

­ найти самую большую цифру числа;

­ n найти сумму цифр числа, больших 5;

­ ответить на вопрос, сколько раз данная цифра встречается в числе? Для этого следует добавить еще одно поле для ввода искомого значения.

**Самостоятельное задание №2.** Решите задачу 1 из практической работы №7 (Вывести таблицу значений функции y=sin(x), если х меняется от 1 до 2 с шагом 0.2) двумя способами с использованием оператора **While** и **Do**.

**Задание 3.** Составить программу перестановки первой и последней цифр введенного натурального числа. Пусть пользователем введено число 4538. После перестановки первой и последней цифр число станет таким: 8534.

Определить последнюю цифру числа нетрудно. Это можно сделать уже известным нам способом: 4538 **%** 10.

Чтобы найти и отделить первую цифру числа, надо использовать прием, который применялся в предыдущей программе для вывода цифр числа и для подсчета суммы цифр, т. е. отделять по одной цифре справа. Но, если в предыдущих программах такой процесс продолжался до тех пор пока **n** не равнялось нулю, а когда n становилось равным нулю, то цикл заканчивался, т. е. все цифры, включая первую, отделялись, то теперь надо этот процесс остановить на одну цифру раньше и тогда последним значением переменной n будет первая цифра числа. В нашем примере она равна **4**.

Итак, первая и последняя цифры найдены. Как переставить их в числе.

Для веденного нами числа это можно сделать так. Вычесть из него первую цифру, умноженную на 1000 и вычесть последнюю цифру: 4538 -4 \* 1000 - 8 = 530.

К полученному результату прибавить последнюю цифру - 8, умноженную на 1000 и прибавить первую цифру: 530 +8 \* 1000 + 4 = 8534.

Две последние операции можно записать в одной строке: 4538 -4 \* 1000 - 8 +8 \* 1000 + 4 = 8534.

Возникает одна трудность. Как определить разряд, в котором находится первая цифра числа (первая слева) и на сколько надо умножить ее при вычитании? Тысячи ли это, десятки тысяч или другой разряд?

Для того, чтобы это выяснять заведем переменную, первоначальное значение которой 1, а затем, каждый раз при отделении цифры она умножается на 10.

Посмотрим весь процесс на примере того же числа 4538.

Первоначальные значения: n = 4538, i = 1.

Цикл продолжается пока n >= 10, 4538 >= 10 - истина, значит операторы цикла выполняются первый раз:

i = i \*10 = 1\*10 = 10; - переменная i получает первое значение, n = 4538 div 10 = 453.

Проверка условия: 453 >= 10 - условие выполняется, значит цикл выполняется второй раз:

i = i \*10 = 10\*10 = 100; n = 453 div 10 = 45.

Проверка условия: 45 >= 10 - истина, значит цикл выполняется третий раз:

i = i\*10 = 100\*10 = 1000;

n := 45 div = 4.

Проверка условия: 4 >= 10 - ложь, значит операторы цикла не выполняются.

Цикл заканчивается.

Конечные значения переменных: n = 4 - первая цифра числа, i = 1000.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Перестановка первой и последней цифры числа**.
* Разместите компоненты **Button, TextBox, Label** как показано на рисунке. В свойстве **Font** установите размер шрифта равным **12** пт. В свойствах **Text** задайте значения, соответствующие рисунку 96.

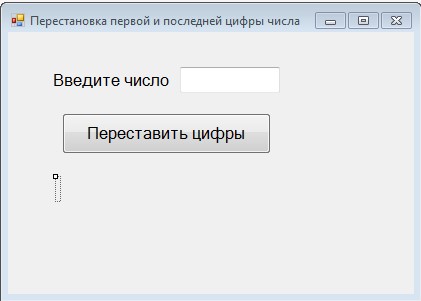


Рисунок 96 – Интерфейс приложения

* Создайте событие для элемента **Button1**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 97). Комментарии вводить не нужно. Разберитесь в коде самостоятельно, используя описание алгоритма в начале задания.

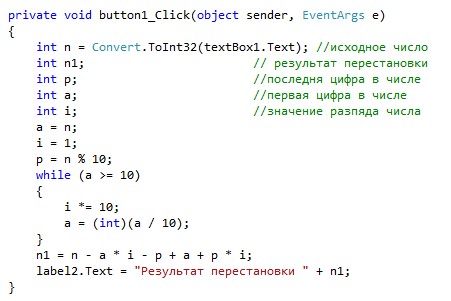


Рисунок 97 – Код события нажатия на кнопку Переставить цифры - Запустите приложение на выполнение и протестируйте программу (Рисунок 98).

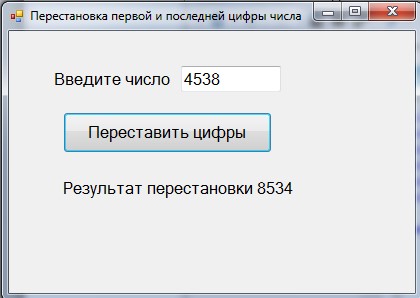


Рисунок 98- Результат тестирования приложения

* Измените программу, используя цикл с постусловием (Рисунок 99).

Запустите приложение на выполнение и протестируйте программу.

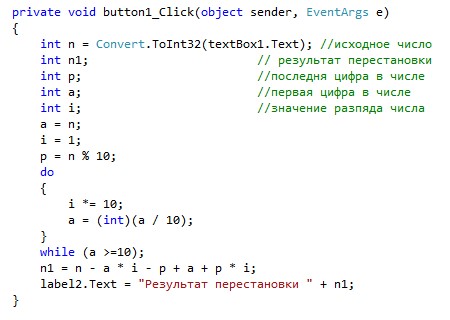


Рисунок 99 – Модифицированный код события

**Самостоятельное задание №3.** Напишите программу для решения одной из предложенных ниже задач.

**Вариант 1.** Составить программу вывода всех натуральных чисел, меньших n, квадрат суммы цифр которых равен заданному числу m.

**Вариант 2.** Известны оценки каждого из 10 учеников класса по физике (использовать случайные числа). Определить количество пятерок, четверок, троек и двоек.

**Вариант 3.** Известна масса каждого человека из некоторой группы людей (15 человек). Людей, имеющих массу более 100 кг, будем условно называть «большими». Определить среднюю массу «больших» людей и среднюю массу всех остальных.

**Вариант 4.** Известен рост каждого человека в группе из 10 человек. На сколько рост самого высокого из них превышает рост самого низкого?

**Вариант 5.** Известны оценки ученика по 12 предметам. Верно ли, что среди них нет троек?

**Вариант 6.** Каждая бактерия делится на две в течение одной минуты.

Начальное значение бактерий равно B. Сколько их станет через A минут.

**Вариант 7.** Стоимость доставки самолетом 1кг груза составляет 30 рублей. Для каждого из N пассажиров введите вес багажа, который он перевозит. Подсчитайте, сколько каждый пассажир должен заплатить за перевозку своего багажа. Вычислите также общий вес груза, перевозимого пассажирами.

**Вариант 8.** Население города увеличивается на 3% каждый год. В 1983 году население города составляло 65000 человек. Напишите программу, которая выведет на экран предсказываемую численность населения города в каждом году, вплоть до 2000.

**Вариант 9.** При изготовлении детали измеряется ее диаметр. Если он больше 4 и меньше 5, то деталь – набракованная. Определите процент брака в партии из N деталей.

**Вариант 10.** В лифте десятиэтажного дома кнопка первого этажа находится на высоте 1м 20см от пола. Кнопка каждого следующего этажа выше предыдущего на 10см. Получить таблицу этажей, до которых сможет доехать маленький мальчик. При этом известно, что рост мальчика N см (случайное число от 90см до 1м50см), а подпрыгивая, он может дотянуться до высоты, превышающей его рост на 45см.

**Вариант 11.** Друзья составили про Петю задачу: наш друг Петя ест невкусную макаронину длиной 60км. Каждый день он съедает N метров этой макаронины (случайное число от 50 до 500). Сколько дней понадобится Пете, чтобы съесть всю макаронину? Сколько метров этой макаронины осталось Пете на последний день? Показать таблицу ежедневных Петиных достижений.

**Вариант 12.** Вы должны своему другу 5 рублей. Вместо того, чтобы попросить вас вернуть ему сразу весь долг, Ваш друг предлагает, чтобы вы платили ему определенную сумму еженедельно в течение 15 недель. Он просит вас уплатить ему в первую неделю 1 копейку, во вторую - 2, в третью - 4, в четвертую - 8 и т.д. Вы уже почти согласились с его предложением, но затем решаете сначала написать программу, чтобы проверить, какую сумму вам придется заплатить, если вы примете его предложение. Напишите эту программу и решите, каким способом лучше выплатить свой долг.

**Вариант 13.** По стеблю растения, высота которого 1 м, от земли ползет гусеница. Днем она поднимается на 3 дм, а ночью опускается на 2 дм. Через сколько суток гусеница доползет до верхушки растения. Выдать таблицу движения гусеницы по дням.

**Вариант 14.** Десять совершенно одинаковых давно нестриженных барана пришли в парикмахерскую. С каждого барана состригли некоторое количество шерсти. Причем минимальное количество состриженной шерсти с одного барана составляет 300 г, а максимальное – 6 кг (в этом случае баран постригся наголо). Выдать таблицу значений, из которой будет видно: сколько шерсти состригли с каждого барана и сколько еще шерсти на нем осталось.

**Вариант 15.** В стаде 40 баранов. Один баран поджал 2 ноги и сразу упал, второй баран поджал 3 ноги и еще быстрее упал, а третий поджал только одну ногу и поэтому не упал. Определить: сколько ног поджали все бараны в стаде, если известно, что каждый баран поджал некоторое количество ног: минимум - одну, максимум – три (получить случайным образом). А также определить: сколько баранов отдыхает на земле и сколько баранов гордо стоит.

**Вариант 16.** Вам необходимо найти цифровой корень числа. Цифровой корень числа вычисляется следующим образом: складываются все цифры числа, затем все цифры найденной суммы и т.д. Процесс повторяется до тех пор, пока в результате не будет получено однозначное число (цифра), которое и является цифровым корнем.

**Вариант 17.** На летних каникулах студент Вася отправился на турбазу «Горная вершина». Ему предстоял поход по маршруту повышенной сложности. Необходимо было преодолеть N горных уступов. При этом спуски и подъемы с уступа и на уступ можно было осуществить только при помощи лестницы. Чтобы подняться с одного горного уступа на другой, Вася должен взять лестницу такой длины, чтобы она была не меньше абсолютной разности высот этих уступов. То же самое происходит и при спуске. Поход Васе предстоит долгий и тяжелый, провизии нужно взять много, поэтому Вася не хочет тащить слишком длинную и тяжелую лестницу. Помогите Васе минимизировать длину лестницы, но так, чтобы он все-таки смог пройти свой нелегкий путь. Считается, что Вася изначально находится на вершине первого уступа, а итоговая цель маршрута – оказаться на вершине самого последнего уступа.

**Критерии оценки:**

**«отлично»**

* правильно решены все задания с 1 по 3;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены все самостоятельные задания №1-3, самостоятельно.

**«хорошо»**

* правильно решены все задания с 1 по 3;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены два самостоятельных задания, с подсказками преподавателя.

**«удовлетворительно»**

* правильно решены все задания с 1 по 3;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.  решено одно самостоятельное задание.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы.

**Практическая работа №9 «Создание проектов для работы с одномерными массивами»** 2 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить заполнение одномерных массивов разными способами.
3. Освоить вывод элементов одномерного массива в элементы ListBox.
4. Освоить выполнение типовых операций обработки одномерных массивов.

**Учебные задачи:**

1. Научиться создавать проекты для работы с одномерными массивами.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Массив представляет собой совокупность данных одного типа с общим для обращения к ним именем. В С# массивы могут быть как одномерными, так и многомерными. Массивы служат самым разным целям, поскольку они предоставляют удобные средства для объединения связанных вместе переменных.

Массивами в С# можно пользоваться практически так же, как и в других языках программирования. Тем не менее, у них имеется одна особенность: они реализованы в виде объектов.

Типичными операциями при работе с массивами являются:

* вывод элементов массива;
* ввод данных в массив;
* определение суммы, произведения и количества каких-либо элементов массива;
* поиск максимального или минимального элемента массива;  поиск заданного элемента массива.

В языке C# каждый индекс изменяется в диапазоне от 0 до некоторого конечного значения.

Объявление одномерного массива выглядит следующим образом: тип[] имя массива;

При объявлении массива используется имя с инициализацией, элементы константного массива следует заключать в фигурные скобки. double[] x = {5.5, 6.6, 7.7};

Создание массива выполняется с помощью операции new. int[] d = new int[5];

Здесь объявлен динамический целочисленный массив, в котором будут храниться 5 целых чисел.

Заполнение данными с клавиатуры массив A.

for(int i = 0; i < 5; i++) A[i] = int.Parse(Console.ReadLine()); Вычисляются элементы массива C.

for(int i = 0; i < 5; i++) C[i] = A[i] + B[i];

**Вопросы для закрепления теоретического материала:**

1. Что такое массив?
2. Как осуществляется доступ к элементам массива?
3. Для чего нужно описывать массивы?
4. Как осуществляется объявление массивов на С#?
5. Что такое размерность массива?
6. Опишите способы задания одномерного массива.
7. Какие способы сортировки массивов вы знаете?

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Заполнить одномерный массив числовыми данными случайным образом (из промежутка [-10; 10]). Вывести массив в **ListBox**. Определить сумму положительных и количество отрицательных элементов массива.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Одномерный массив**.
* Разместите компоненты **Button, Label, ListBox** как показано на рисунке

100. В свойстве **Font** установите размер шрифта равным **12** пт. Удалите текст из компонента **Label1** и **Label2.**

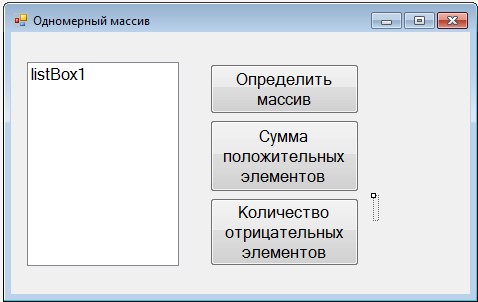
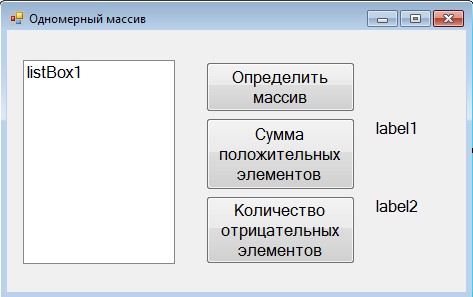


Рисунок 100 - Интерфейс приложения

* Создайте событие для элемента **Button1 (Определить массив)**, для этого достаточно дважды щелкнуть мышью по данному элементу. Объявите переменную массива выше процедуры, как показано на рисунке 101.

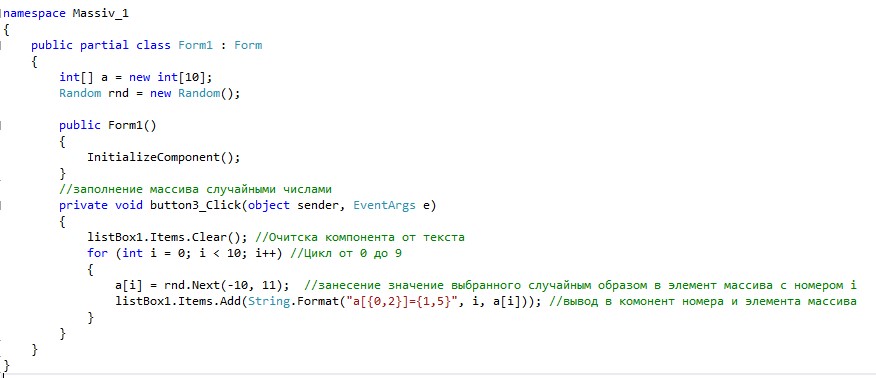


Рисунок 101 – Код события нажатия на кнопку Определить массив

* Создайте событие для элемента **Button2 (Сумма положительных элементов)**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 102).

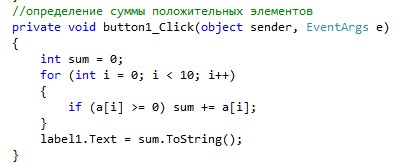


Рисунок 102 – Код события нажатия на кнопку Сумма положительных элементов

* Создайте событие для элемента **Button3 (Количество отрицательных элементов)**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 103).

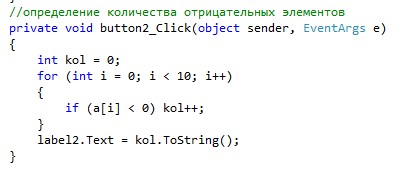


Рисунок 103 – Код события нажатия на кнопку Количество отрицательных элементов

* Запустите приложение на выполнение и протестируйте программу (Рисунок 104).

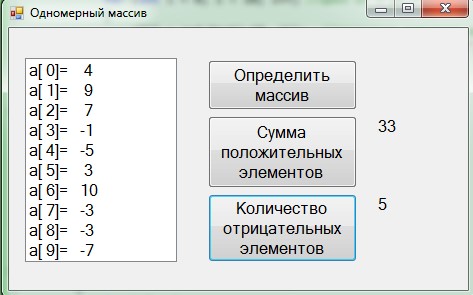


Рисунок 104- Результат тестирования приложения

**Самостоятельное задание №1.** Модифицируйте программу в задании 1, добавив кнопку и изменив диапазон случайных чисел, в соответствии с вариантом (таблица 10). Если требуется разместите необходимые компоненты на форме.

Таблица 10- Варианты заданий модификации

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Диапазон значений** | **Задание** |
| 1 | -100 - 100 | Заменить все элементы с отрицательным значением на значение минимального не равного 0 положительного элемента |
| 2 | -50 - 50 | Подсчет количества элементов массива кратных 3. |
| 3 | 0 - 100 | Подсчет количества элементов массива с нечетными индексами. |
| 4 | -50 - 50 | Подсчет количества элементов массива, лежащие в интервале от с1 до с2. |
| 5 | -100 - 100 | Подсчет суммы элементов массива кратных 3. |
| 6 | -100 - 100 | Подсчет суммы элементов массива с нечетными индексами. |
| 7 | -100 - 100 | Подсчет суммы элементов массива с четными индексами. |
| 8 | -100 - 100 | Заменить все элементы с отрицательными значениями средним арифметическим значением всех элементов массива |
| 9 | 0 - 100 | Подсчет суммы элементов массива, лежащие в интервале от с1 до с2. |
| 10 | 0 - 100 | Подсчет произведения элементов массива кратных 3. |
| 11 | -50 - 50 | Подсчет произведения элементов массива с нечетными индексами. |
| 12 | -100 - 100 | Подсчет произведения элементов массива, лежащие в интервале от с1 до с2. |
| 13 | 0 - 100 | Обменять местами наибольший и наименьший элементы. |
| 14 | -50 - 50 | Определить, сколько среди элементов массива чисел меньших K, равных K и больших K. |
| 15 | -100 - 100 | Подсчитать, сколько в массиве отрицательных, положительных и нулевых элементов. |

**Самостоятельное задание №2.** Составьте проект для решения задачи по варианту. Если сказано, что массив заполнен некоторыми числами – используйте либо заполнение случайными числами (диапазон случайных чисел определяйте самостоятельно, опираясь на смысл задачи), либо заполнение с клавиатуры.

**Вариант 1.** В массив из 7 элементов занесены натуральные числа. Найти сумму тех элементов, которые кратны К.

**Вариант 2.** Дана последовательность десяти действительных чисел. Заменить все ее члены, большие данного числа, самим эти числом. Подсчитать количество замен.

**Вариант 3.** Записать в массив Y(5) квадраты чисел, указанных в массиве N(5).

**Вариант 4.** Вывести на печать первый отрицательный элемент массива A(20) и его номер, полагая, что в массиве есть хотя бы один отрицательный элемент.

**Вариант 5.** Дана последовательность действительных чисел. Подсчитать, сколько в этой последовательности положительных, отрицательных и нулевых элементов.

**Вариант 6.** Определить количество элементов последовательности натуральных чисел, кратных некоторому числу и заключенных в определенном промежутке.

**Вариант 7.** Даны целые числа. Вывести на экран только те числа, значение которых больше их индекса.

**Вариант 8.** Найти разность максимального и минимального элементов массива X(10).

**Вариант 9.** Дан массив из 10 элементов. Подсчитайте количество таких элементов:

* которые больше среднего арифметического всех его элементов;  которые равны максимальному значению этого массива.

**Вариант 10.**  Пусть дана последовательность из 15 различных целых чисел. Найдите среднее арифметическое чисел этой последовательности, расположенных между максимальным и минимальным числами.

**Вариант 11.**  Дан набор из 10 целых чисел. Вывести в том же порядке все четные (нечетные) числа из данного набора и количество K таких чисел.

**Вариант 12.**  Выяснить, верно ли, что произведение элементов массива есть неотрицательное число.

**Вариант 13.** Дан массив. Все его элементы:  увеличить в 2 раза;

* уменьшить на число a.

**Вариант 14.** Дан массив. Определить:

* сумму всех элементов массива;
* сумму первых шести элементов массива;
* сумму элементов массива с k1-го по k2-й (k1 и k2 вводятся с клавиатуры);

**Вариант 15.** Дан массив. Определить:  произведение всех элементов массива;

* сумму квадратов всех элементов массива;
* среднее арифметическое элементов массива с p1-го по p2-й (p1 и p2 вводятся с клавиатуры).

**Вариант 16.** Дан массив целых чисел. Выяснить:

* верно ли, что сумма элементов массива есть четное число;
* верно ли, что сумма квадратов элементов массива есть пятизначное число.

**Критерии оценки:**

**«отлично»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены все самостоятельные задания №1-2, самостоятельно.

**«хорошо»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание 2.

**«удовлетворительно»**

* правильно решено задание 1;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* решено самостоятельное задание 1.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы.

**Практическая работа №10 «Создание проектов для работы с двумерными массивами»**

2 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить заполнение двумерных массивов разными способами.
3. Освоить вывод элементов двумерного массива в сетку DataGridView.
4. Освоить выполнение типовых операций обработки двумерных массивов. **Учебные задачи:**

1. Научиться создавать проекты для работы с двумерными массивами.

**Образовательные результаты, заявленные во ФГОС ТОП-50:**

Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

В повседневной жизни довольно часто приходится иметь дело с информацией, которая представлена в табличной форме. Колонки и (или) строки таблицы, как правило, состоят из однородной информации. Поэтому в программе, обрабатывающей табличные данные, имеет смысл использовать массивы для хранения и обработки таблиц. Такая таблица может быть представлена как двумерный массив (Рисунок 105).

Для работы с двумерными массивами используются вложенные циклы.

|  |  |  |  |
| --- | --- | --- | --- |
| А11 | А12 | А13 | А14 |
| А21 | А22 | А23 | А24 |
| А31 | А32 | А33 | А34 |

A 3, 4

Рисунок 105 – Одномерный массив

Объявление двумерного массива integer размерами 10x20: int[ , ] table = new int[10, 20];

Оба размера массива разделяются запятой. В первой части этого объявления синтаксическое обозначение [ , ] означает, что создается переменная ссылки на двумерный массив. Если же память распределяется для массива с помощью оператора new, то используется следующее синтаксическое обозначение: int[10, 20]. В данном объявлении создается массив размерами 10x20, но и в этом случае его размеры разделяются запятой.

Для доступа к элементу двумерного массива следует указать оба индекса, разделив их запятой. Например, в следующей строке кода элементу массива table с координатами местоположения (3,5) присваивается значение 10: table[3, 5] = 10;

**Вопросы для закрепления теоретического материала:**

1. Что такое массив?
2. В чем разница между одномерными и двумерными массивами?
3. Как осуществляется описание двумерных массивов на С#?
4. Как можно осуществить ввод матрицы? 5 Как можно организовать вывод матрицы?

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Создайте приложение, которое выводит двухмерный массив случайных чисел в объекте **DataGridView** и определяет минимальный элемент.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Двумерный массив**.
* Разместите компоненты **Button, DataGridView** как показано на рисунке 106. В свойстве **Font** установите размер шрифта равным **12** пт.

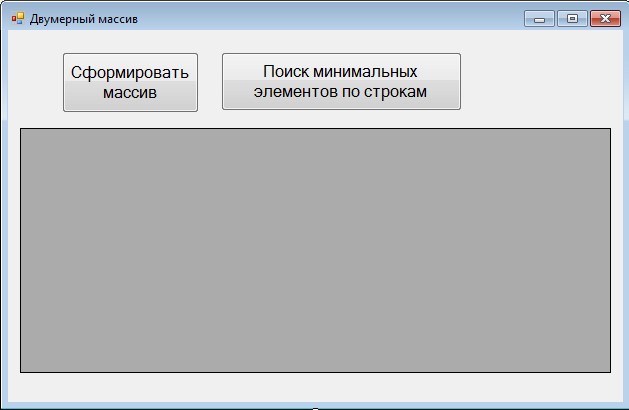


Рисунок 106 – Интерфейс приложения

* Создайте событие для элемента **Button1(Сформировать массив)**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 107).

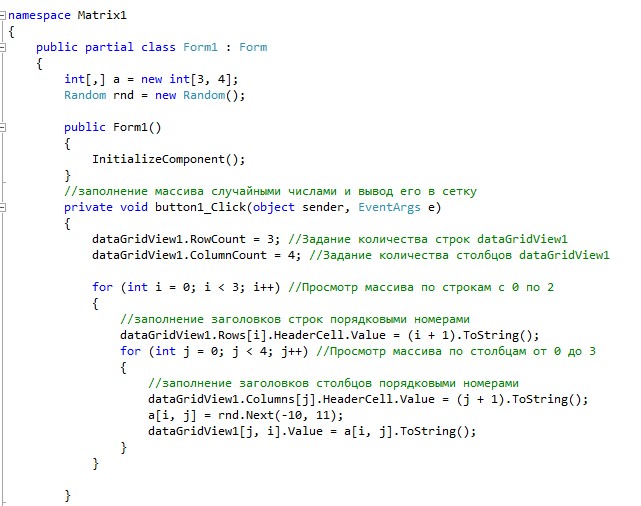


Рисунок 107 – Код события нажатия на кнопку Сформировать массив

* Создайте событие для элемента **Button2(Поиск минимальных элементов по строкам)**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 108).

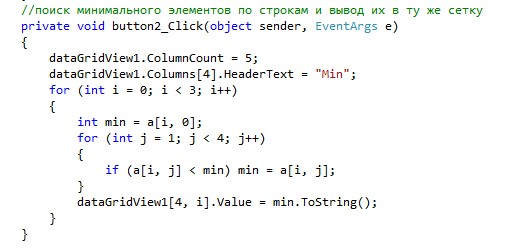


Рисунок 108 – Код события нажатия на кнопку Поиск минимальных элементов по строкам

* Запустите приложение на выполнение и протестируйте программу (Рисунок 109).



Рисунок 109 - Результат тестирования приложения

**Самостоятельное задание №1.** Составьте проект для решения задачи по вариантам. Если сказано, что массив заполнен некоторыми числами – используйте либо заполнение случайными числами (диапазон случайных чисел определяйте самостоятельно, опираясь на смысл задачи), либо заполнение с клавиатуры.

**Вариант 1.** Сформировать двумерный массив M размерностью 4х3 по правилу: элементы первой строки=1, второй строки=2, все остальные=3.

**Вариант 2.** Найти сумму и произведение всех элементов, а также максимальный элемент массива.

**Вариант 3.** Найти сумму всех положительных элементов массива Y[4, 4], найти среднее арифметическое всех элементов. Массив сформировать случайным образом.

**Вариант 4.** Задан двумерный массив из 12-ти элементов. Разделить каждый элемент массива на 7. Результат сохранить в том же массиве.

**Вариант 5.** Дан двумерный массив. Определить:

* сумму элементов второго столбца массива, больших 10;
* количество ненулевых элементов первой строки массива;  сумму отрицательных элементов пятой строки массива.

**Вариант 6.** Дан двумерный массив. Определить:

* сумму элементов третьей строки массива, не превышающих 25;  количество элементов второго столбца массива, больших 15;  сумму элементов четвертого столбца массива, меньших 100.

**Вариант 7.** Дан двумерный массив. Определить:

* количество элементов четвертой строки массива, меньших 5;
* среднее арифметическое четных элементов третьего столбца;  среднее арифметическое элементов четвертой строки, кратных трем.

**Вариант 8.** Дан двумерный массив. Определить:

* количество ненулевых элементов второго столбца массива, больших 15;
* среднее арифметическое нечетных элементов пятой строки;
* среднее арифметическое элементов четвертого столбца, кратного четырем.

**Вариант 9.** В массиве записаны целые числа. Вычислить:

* произведение элементов побочной диагонали массива, меньших 10;  сумму элементов главной диагонали массива, оканчивающихся цифрой 7. **Вариант 10.** В массиве записаны целые числа. Вычислить:
* количество нулевых элементов, расположенных над главной диагональю массива;  число нечетных элементов, расположенных под побочной диагональю массива.

**Вариант 11.** Дана матрица размера *M* ×*N* и целое число *K* (1 ≤*K* ≤*M*). Найти сумму и произведение элементов *K*-й строки данной матрицы.

**Вариант 12.** Дана матрица размера *M* ×*N* и целое число *K* (1 ≤*K* ≤*M*). Найти сумму и произведение элементов *K*-го столбца данной матрицы.

**Вариант 13.** В двумерном массиве хранится информация о количестве студентов в той или иной группе каждого курса института с первого по пятый (в первой строке — информация о группах первого курса, во второй — второго и т.

д.). На каждом курсе имеется 8 групп. Составить программу для расчета общего числа студентов на любом курсе.

**Вариант 14.** В двухмерном массиве хранится информация о зарплате 5 человек за каждый месяц года (первого человека — в первой строке, второго — во второй и т. д.). Составить программу для расчета общей зарплаты, полученной за год любым человеком, информация о зарплате которого представлена в массиве.

**Вариант 15.** В поезде 10 вагонов, в каждом из которых 36 мест. Информация о проданных на поезд билетах хранится в двумерном массиве, номера строк которых соответствуют номерам вагонов, а номера столбцов — номерам мест. Если билет на то или иное место продан, то соответствующий элемент массива имеет значение 1, в противном случае — 0. Составить программу, определяющую число свободных мест в любом из вагонов поезда.

**Вариант 16.** Дан двумерный массив целых чисел. В каждой его строке (столбце) найти:

* сумму отрицательных элементов;
* среднее арифметическое отрицательных элементов;

**Вариант 17.** Дан двумерный массив целых чисел. В каждой его строке (столбце) найти:

* количество четных элементов;
* среднее арифметическое положительных элементов, кратных пяти; **Вариант 18.** Дан двумерный массив целых чисел. В каждой его строке (столбце) найти:
* количество элементов, попадающих в промежуток от a до b;
* сумму нечетных положительных элементов;

**Вариант 19.** Дан двумерный массив целых чисел. В каждой его строке (столбце) найти:

* количество положительных элементов, кратных k;
* сумму элементов в строках с k1 по k2;

**Вариант 20.** Дан двумерный массив целых чисел. В каждой его строке (столбце) найти:

* количество элементов, кратных a или b;
* произведение отрицательных элементов в столбцах с k1 по k2;

**Критерии оценки:**

**«отлично»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, самостоятельно и вариант соответствует номеру ПК.

**«хорошо»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание 1с недочетами и вариант соответствует номеру ПК.

**«удовлетворительно»**

* правильно решено задание 1;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* решено самостоятельное задание, но вариант не соответствует номеру ПК.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической ра-

боты

**Тема 3 «Объектно-ориентированная модель программирования»**

**Практическая работа №11 «Создание простейших методов»**

2 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить создание методов, не возвращающих значения.
3. Освоить использование методов в проекте. **Учебные задачи:**

1. Научиться создавать простые методы и использовать их в своей программе.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
* Подпрограммы, составление библиотек подпрограмм

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

В языке C# нет специальных ключевых слов - procedure и function, но присутствуют сами эти понятия.

Синтаксис объявления метода позволяет однозначно определить, чем является метод - процедурой или функцией.

Функция отличается от процедуры двумя особенностями:

1. Она всегда вычисляет некоторое значение, возвращаемое в качестве результата функции;
2. И вызывается в выражениях.

Процедура C# имеет свои особенности:

1. Она возвращает формальный результат void, указывающий на отсутствие результата;
2. Вызов процедуры является оператором языка;
3. И она имеет входные и выходные аргументы, причем выходных аргументов - ее результатов - может быть достаточно много.

Обычно метод предпочитают реализовать в виде функции тогда, когда он имеет один выходной аргумент, рассматриваемый как результат вычисления значения функции. Возможность вызова функций в выражениях также влияет на выбор в пользу реализации метода в виде функции. В других случаях метод реализуют в виде процедуры.

Написание метода всегда состоит из двух этапов: подготовка кода решаемого алгоритма и определение входных и выходных параметров. Методы могут работать как с параметрами, так и без них. Если метод не возвращает никаких значений, то к его заголовку добавляется ключевое слово **void**. Методы целесообразно помещать в отдельные классы.

Статические поля – поля, принадлежащие классу. Они объявляются с ключевым словом **static**. Основное отличие от обычных полей – для обращения к статическим полям не требуется создание объекта. Доступ осуществляется напрямую через имя класса. Более того, через объекты к статическим полям обратиться нельзя.

Пример объявления статического поля: **public static int I;** Синтаксис заголовка метода:

[атрибуты][модификаторы доступа]{void| тип\_результата\_функции} имя\_метода([список\_формальных\_аргументов])

Квадратные скобки в формуле показывают, что атрибуты и модификаторы могут быть опущены при описании метода.

Модификаторы доступа:

* **public** показывает, что метод открыт и доступен для вызова клиентами и потомками класса.
* **private** говорит, что метод предназначен для внутреннего использования в классе и доступен для вызова только в теле методов самого класса.

Если модификатор доступа опущен, то по умолчанию предполагается, что он имеет значение **private** и метод является закрытым для клиентов и потомков класса.

Обязательным при описании заголовка является указание типа результата, имени метода и круглых скобок, наличие которых необходимо и в том случае, если сам список формальных аргументов отсутствует. Формально тип результата метода указывается всегда, но значение **void** однозначно определяет, что метод реализуется процедурой. Тип результата, отличный от **void**, указывает на функцию. Вот несколько простейших примеров описания методов:

void A() {...}; int B(){...}; public void C(){...};

Методы A и B являются закрытыми, а метод С - открыт. Методы A и С реализованы процедурами, а метод B - функцией, возвращающей целое значение.

Вызов метода, независимо от того, процедура это или функция, имеет один и тот же синтаксис:

имя\_метода([список\_фактических\_аргументов])

Если это оператор, то вызов завершается точкой с запятой. Формальный аргумент, задаваемый при описании метода - это всегда имя аргумента (идентификатор). Фактический аргумент - это выражение, значительно более сложная синтаксическая конструкция.

**Вопросы для закрепления теоретического материала:**

1. Что такое метод?
2. Назовите модификаторы доступа.
3. Какое ключевое слово необходимо использовать в заголовке метода, если метод не возвращает значений?
4. Перечислите основные моменты, которые необходимо соблюдать при вызове метода.
5. С какой целью используется ключевое слово static?

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Известны катеты нескольких прямоугольных треугольников. Определить для каждого треугольника значение гипотенузы.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Гипотенуза**.
* Разместите компоненты **Button, Label, numericUpDown** как показано на рисунке. В свойстве **Font** установите размер шрифта равным **12** пт. В свойствах **Text** задайте значения, соответствующие рисунку 110.

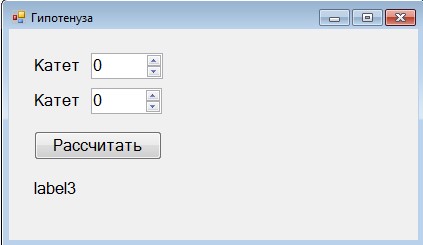


Рисунок 110 – Интерфейс приложения

* Создайте обработчик событий на кнопку **Рассчитать** (Рисунок 111).



Рисунок 111 – Код события нажатия на кнопку Рссчитать

Компилятор подчеркнул метод **Hip**, т. к. не может его найти не в одном из встроенных классов.

* Напишите метод **Hip**, который будет высчитывать значение гипотенузы (Рисунок 112). Создаем метод ниже события нажатия на кнопку, т. е. после закрывающийся **}**.

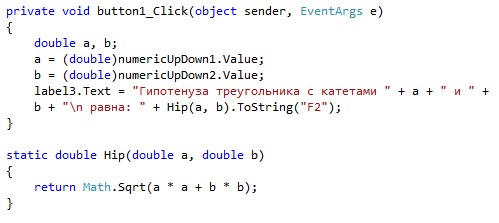


Рисунок 112 – Метод вычисления значения гипотенузы

Данный метод мы вызвали из нажатия кнопки сразу по имени, передав только параметры. Создавать объект не потребовалось, так как перед методом указали слово **static** (статический).

* Запустите программу на выполнение. Проверьте работоспособность (Рисунок 113).

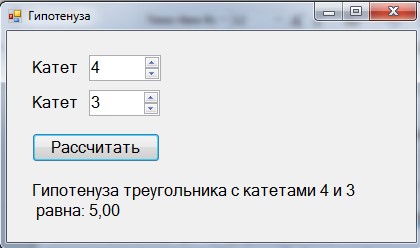


Рисунок 113- Результат тестирования приложения

**Самостоятельное задание № 1.** Создайте новый проект для решения следующей задачи с использованием методов по вариантам.

**Вариант 1.** Даны две переменные x, y (вещественные). Используя метод нахождения наименьшего из двух значений, найти наименьшее из x, y.

**Вариант 2.** Даны две переменные x, y (вещественные). Используя метод нахождения наименьшего из двух значений, найти наименьшее из x+y, x∙y, 0.5. **Вариант 3.** Даны две переменные x, y (вещественные). Используя метод нахождения наименьшего из двух значений, найти наименьшее из 2x, |x-y|, 4.5, (x+y)/3.

**Вариант 4.** Разработать метод max(a,b) для нахождения максимального из двух чисел. Вычислить с помощью него значение выражения z=max(x,2yx)+max(5x+3y,y).

**Вариант 5.** Разработать метод f(n), который для заданного натурального числа n находит значение *n+n*. Вычислить с помощью него значение выражения:

**Вариант 6.**

2

21

21

2

13

13

2

6

6

*+*

*+*

*+*

*+*

*+*

.

Описать метод, находящий величину угла в радианах, если

дана его величина D в градусах (D — вещественное число, 0 ≤ D ≤ 360). Воспользоваться следующим соотношением: 180◦ = π радианов.

**Вариант 7.** √𝑝 ∙ (𝑝 − 𝑎) ∙ (𝑝 − 𝑏) ∙ (𝑝 − 𝑐)Описать метод, находящий величину угла в градусах, если дана его величина D в радианах (D — вещественное число). Воспользоваться следующим соотношением: 180◦ = π радианов. Методы: вычисление длины отрезка, вычисление площади по формуле Герона

𝑎+𝑏+𝑐

𝑝 = е

2

**Вариант 8.** Даны координаты трех вершин треугольника. Найти длины всех его сторон и периметр, если треугольник существует.

√𝑝 ∙ (𝑝 − 𝑎) ∙ (𝑝 − 𝑏) ∙ (𝑝 − 𝑐)Используйте метод нахождения длины отрезка, заданного координатами своих концов. Методы: вычисление длины отрезка, вычисление площади по формуле Герона

𝑎+𝑏+𝑐

𝑝 =

2

**Вариант 9.** Напишите программу поиска большего из четырех чисел с использованием метода поиска большего из двух чисел.

**Вариант 10.** Составьте программу для вычисления длин медиан треугольника, у которого длины сторон a, b, c. Длина медианы будет рассчитываться в методе.

**Вариант 11.** Даны координаты двух точек A(x1,y1) и B(x2,y2) в прямоугольной системе координат. Какая из этих точек находится дальше от начала координат, определить с помощью метода.

**Вариант 12.** Даны три числа x, y, z. Найти значение выражения max (x+y+z, xyz)+3, с помощью метода.

**Критерии оценки:**

**«отлично»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, самостоятельно.

**«хорошо»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, с ошибками.

**«удовлетворительно»**

* правильно решено задание 1;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* не решено самостоятельное задание №1.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы

**Практическая работа №12 «Создание простейших классов»**

6 часов

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить алгоритм построения класса.
3. Освоить создание объектов класса. **Учебные задачи:**

1. Научиться создавать проекты с использованием простых классов.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
* 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.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Все программы на языке C# создаются внутри классов. Основная структура класса имеет следующий вид:



Слово **class** должно быть написано строчными буквами, но имя класса может быть любым и содержать как прописные, так и строчные буквы, причем использование пробелов не допускается.

**Класс** представляет собой шаблон, по которому определяется форма объекта. В нем указываются данные и код, который будет оперировать этими данными. В С# используется спецификация класса для построения объектов, которые являются экземплярами класса. При этом очень важно подчеркнуть, что класс является логической абстракцией. Физическое представление класса появится в оперативной памяти лишь после того, как будет создан объект этого класса.

При определении класса объявляются данные, которые он содержит, а также код, оперирующий этими данными. Если самые простые классы могут содержать только код или только данные, то большинство настоящих классов содержит и то, и другое.

Объект можно рассматривать как совокупность свойств, методов и событий, работающий при этом как единое целое. Синтаксис класса: class имя\_класса{ тип\_доступа тип имя\_переменной1; тип\_доступа тип имя\_переменной2;

…

тип\_доступа возвращаемый\_тип

имя\_метода1(список\_параметров) {тело\_метода}

}

где тип\_доступа может быть **public, private, protected, internal**. Члены класса с типом доступа **public** доступны везде за пределами данного класса, с типом доступа **protected** – внутри членов данного класса и производных, с типом доступа **private** - только для других членов данного класса. Тип доступа **internal** применяется для типов, доступных в пределах одной сборки. Пример: class Animal

{ public string Name; private int Weight; protected int Type;

public int Animal(int W, int T, string N)

{

Weight=W;

Type=T;

Name=N;

}

public int GetWeight()

{

return Weight;

}

}

Создание объекта:

имя\_класса имя\_обьекта = new имя\_класса();

При создании объекта класса происходит вызов соответствующего конструктора класса.

**Вопросы для закрепления теоретического материала:**

1. Что такое класс?
2. Что такое объект?
3. Назовите основные принципы ООП и дайте каждому из них характеристику.
4. Какие элементы определяются в составе класса?
5. Каково соотношение понятий «класс» и «объект»? 6 Что понимается под термином «члены класса»?
6. Какие модификаторы типа доступа Вам известны?
7. В чем заключаются особенности доступа членов класса с модификатором public?
8. В чем заключаются особенности доступа членов класса с модификатором private?
9. В чем заключаются особенности доступа членов класса с модификатором protected?
10. В чем заключаются особенности доступа членов класса с модификатором internal?
11. Какое ключевое слово языка C# используется при создании объекта?
12. Приведите синтаксис создания объекта в общем виде.

**Задания и инструкция по выполнению практической работы Задание 1.** Создать простой класс **Worker**.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Работа с классом Рабочий**.
* Разместите компоненты **Button, TextBox, Label, numericUpDown** как показано на рисунке. В свойстве **Font** установите размер шрифта равным **12** пт. В свойствах **Text** задайте значения, соответствующие рисунку 114.

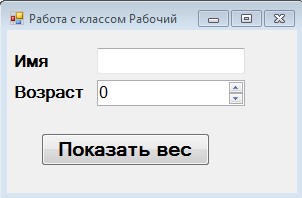


Рисунок 114 – Интерфейс приложения

* В обозревателе решений нажмите ПКМ и в контекстном меню выберите команду **Добавить** – **класс** (Рисунок 115).

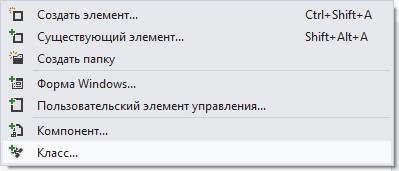
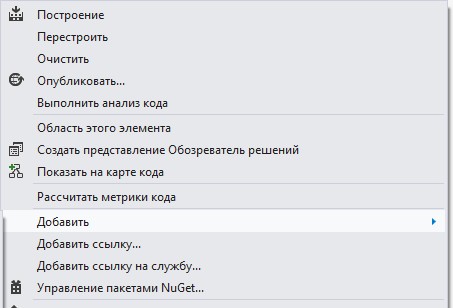


Рисунок 115 – Добавление класса к проекту

* В окне добавления нового элемента задайте имя класса **Worker** и нажмите кнопку **Добавить** (Рисунок 116).

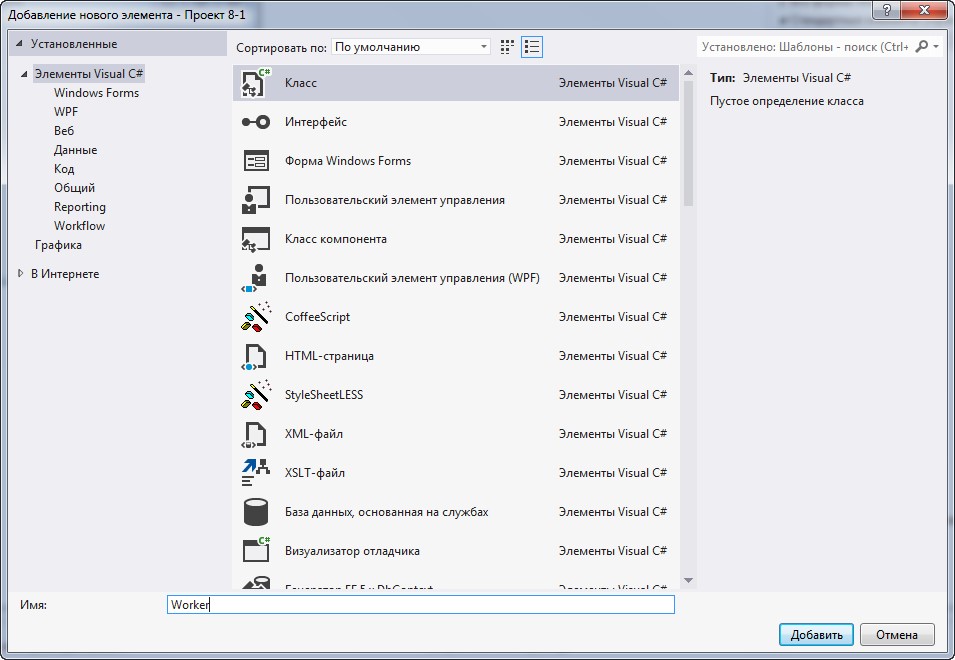


Рисунок 116 – Окно добавления нового элемента

* В результате в код проекта добавиться заготовка для создания класса **Worker** (Рисунок 117).

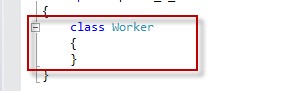


Рисунок 117 – Код создания класса

* В класс добавьте два общедоступных поля: **имя** и **возраст**, а также одно защищённое поле: **вес** (Рисунок 118).

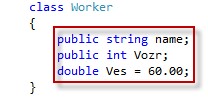


Рисунок 118 – Код создания класса с полями

* Для записи и чтения данных из защищенных полей используйте методы. Добавьте в класс **Worker** метод **SetEat**, который будет отвечать за еду: если человек что-то съест, то его вес должен будет увеличиться на количество съеденного (Рисунок 119). Если поле вес защищенное, то в него не только писать нельзя, но и читать тоже нельзя. Для чтения данных из защищенного поля необходимо использовать еще один метод **GetEat**.

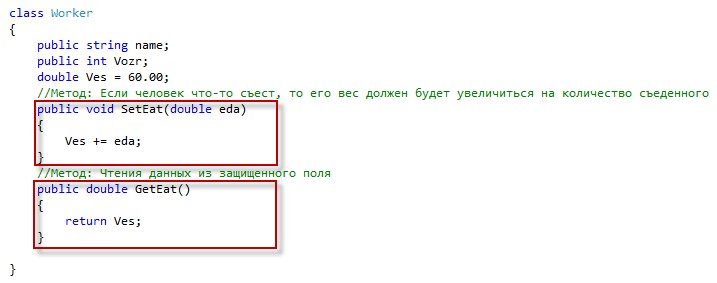


Рисунок 119 – Добавление в класс методов

* Создайте обработчик событий на кнопку **Показать вес** (Рисунок 120), установив значение параметра **eda** сначала 5 кг, затем 15 кг пищи, а потом проверьте его вес.

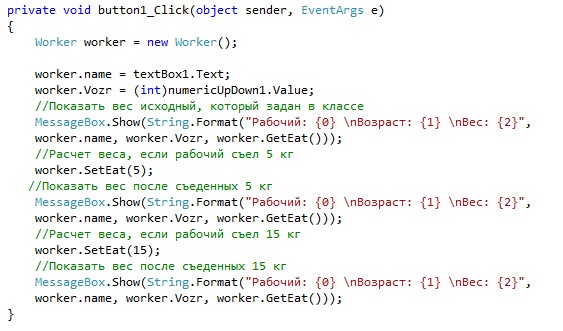


Рисунок 120 – Код события нажатия на кнопку Показать вес

* Запустите программу на выполнение. Проверьте работоспособность (Рисунок 121).

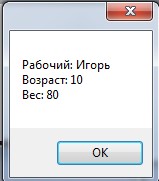
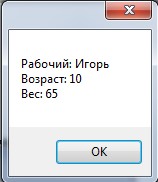
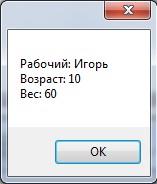
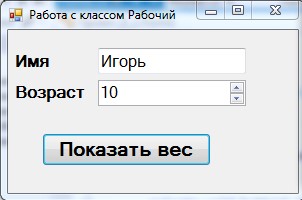


Рисунок 121- Результат тестирования приложения

* Усовершенствуйте метод **SetEat** таким образом, что если рабочий за раз съедает более, чем 10 кг, то его возраст увеличивается на год, а вес увеличивается только наполовину от съеденного.
* Измените данный метод в классе, в соответствии с рисунком 122.

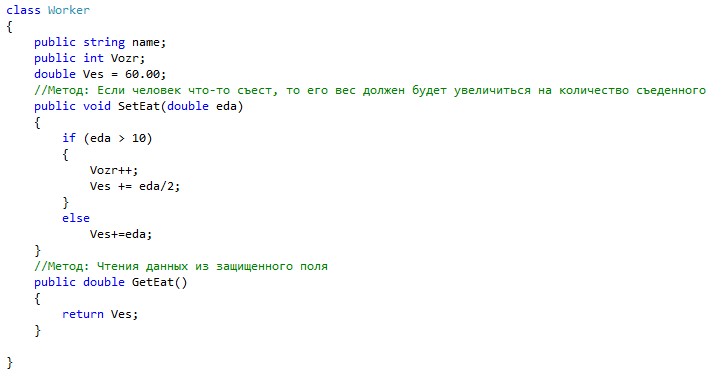


Рисунок 122 – Изменения метода в классе

* Запустите программу на выполнение. Проверьте работоспособность (Рисунок 123). Сохраните и не удаляйте, т. к. она понадобиться в следующей практической работе.

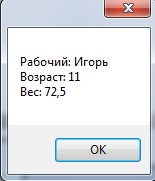
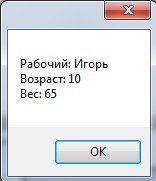
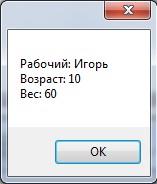


Рисунок 123- Результат тестирования приложения - Обозреватель решений изображён на рисунке 124.

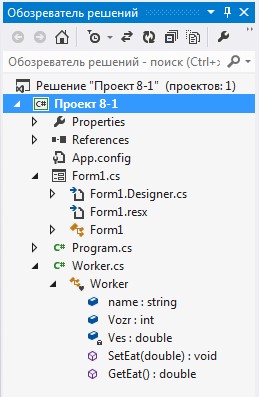


Рисунок 124- Обозреватель решения

**Самостоятельное задание №1.** Создайте проект, в котором можно было бы вводить и получать информацию об определенных объектах некоторого класса:

**Вариант 1.** класс «Книги»: название, автор, год выпуска, объем в листах.

**Вариант 2.** класс «Насекомые»: название, скорость передвижения, окраска, используемая пища, среда обитания.

**Вариант 3.** класс «Газеты»: название, номер, дата выпуска.

**Вариант 4.** класс «Пирожные»: название, тесто, крем, вес, калорийность, цена.

**Вариант 5.** класс «Молочные изделия»: название, вес, жирность, производитель, цена.

**Вариант 6.** класс «Поезд»: номер, пункт отправления, пункт назначения, дата отправления, время отправления, время в пути.

**Вариант 7.** класс «Собака»: кличка, порода, рост по холке, окрас шерсти, вес, характер.

**Вариант 8.** класс «Самолет»: марка самолета, пункт отправления, пункт назначения, дата отправления, время отправления, время в пути.

**Вариант 9.** класс «Студент»: имя, отчество, фамилия, пол, группа, дата рождения, рост, вес, вид спорта.

**Вариант 10.** класс «Телефонный звонок»: номер телефона, дата разговора, продолжительность, код города.

**Вариант 11.** класс «Поездка»: номер поезда, пункт назначения, дни следования, время прибытия, время стоянки.

**Вариант 12.** класс «Кинотеатр»: название кинофильма, сеанс, стоимость билета, зал, количество зрителей.

**Вариант 13.** класс «Товар»: название, цена, единицы измерения, вид товара, количество.

**Вариант 14.** класс «Автомобиль»: арка, мощность, стоимость.

**Вариант 15.** класс «Экзамен»: имя студента, предмет, дата экзамена, оценка.

**Задание 2.** Создайте класс Монтср с защитным полем уровнем жизни и методом поиска уровня жизни после игры. В зависимости от выпавшего случайного значения, тот или иной монстр теряет половину своих сил.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Игра**.
* Разместите компоненты **Button, Label** как показано на рисунке. В свойстве **Font** установите размер шрифта равным **12** пт. В свойствах **Text** задайте значения, соответствующие рисунку 125.

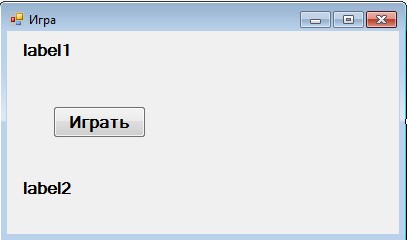


Рисунок 125 – Интерфейс приложения

* В обозревателе решений нажмите ПКМ и в контекстном меню выберите команду **Добавить** – **класс** (рисунок 126).

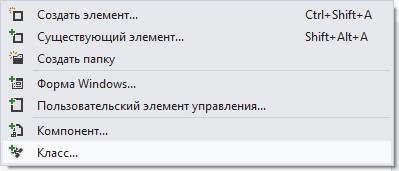
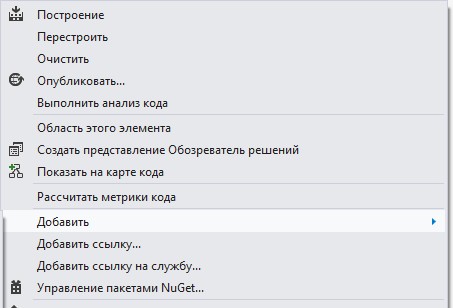


Рисунок 126 – Контексное меню добавления класса

* В окне добавления нового элемента задайте имя класса **Monstr** и нажмите кнопку **Добавить**.
* В результате в код проекта добавиться заготовка для создания класса **Monstr** (рисунок 127).

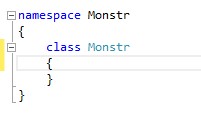


Рисунок 127 – Заготовка для создания класса

* В класс добавьте защищенное поле: **уровень жизни** (Рисунок 128).

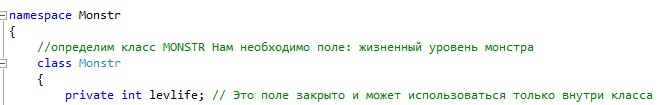


Рисунок 128 – Добавление закрытого поля в класс

* Для того чтобы данное поле было доступно в основной программе следует написать метод на чтение данного поля (Рисунок 129).

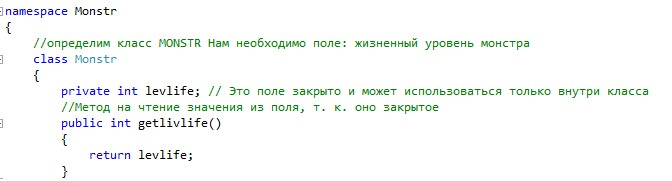


Рисунок 129 – Добавление метода чтения

* Добавьте в класс **Monstr** метод **Min**, который будет отвечать за сокращение уровня жизни монстра в два раза после игры. В качестве входного параметра (а) будет задаваться значение текущего уровня жизни монстра (Рисунок 130).

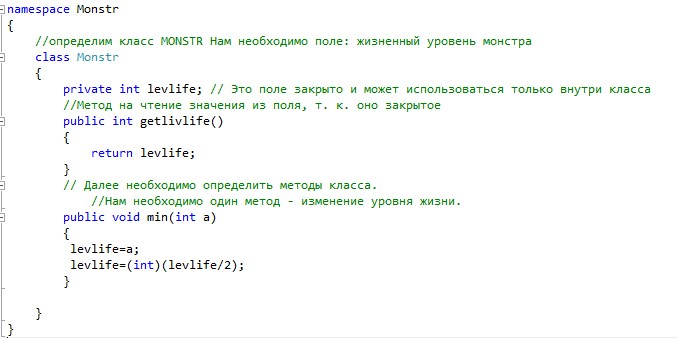


Рисунок 130 – Добавление метода

* Создайте обработчик событий на кнопку **Играть**. Количество жизни в игре у монстра (b) будет задаваться случайным образом (Рисунок 131).

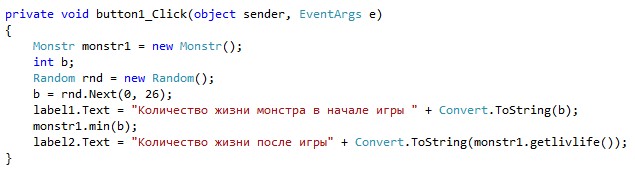


Рисунок 131 – Обработчик события кнопки Играть

* Запустите программу на выполнение. Проверьте работоспособность (Рисунок 132).

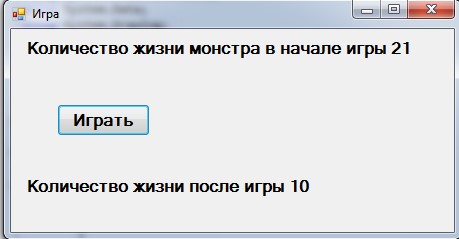


Рисунок 132 - Результат тестирования приложения **Задание 3.** Ответьте на следующие вопросы преподавателю:

* Раскройте последовательность работы с классами;
* Укажите область описания класса;
* Укажите область описания закрытых полей и функций класса;
* Что означает – закрытое поле;
* Укажите область описания открытых полей и функций;
* Что значит открытые поля и функции класса;
* Опишите синтаксис описания класса (заголовок, тело);
* Опишите синтаксис реализации функций класса;
* Опишите технологию вызова функций в теле главной функции.

**Задание 4.** Измените текст программы так, чтобы поле levlife стало доступным в основной функции и значение этого поля задавалось с клавиатуры.

**Задание 5.** Модифицируйте текст программы так, чтобы в системе существовало три монстра с различными уровнями жизни, задание которых осуществляется с клавиатуры.

**Задание 6.** Дополните текст программы описанием функции, которая позволит при уменьшении значения поля levlife определенного монстра, увеличить значение этого поля остальных монстров на 30 очков.

**Самостоятельное задание №2.** Создайте новый проект для работы с приведенным ниже классом:

**Вариант 1.** Класс «Пирожное»:

*Поля класса:* название, тесто, крем, вес, калорийность, цена.

Методы класса:

* вывод информации о пирожном;
* подсчет стоимости покупки пирожных конкретного вида.
* определите стоимость покупки пирожных, если эти пирожные могут быть разного вида.
* дополнительно придумайте к каждому классу еще по одному методу, возвращающему значение.

**Вариант 2.** Класс «Цветы»

Поля класса: название, тип, оттенок, страна-производитель, цена, длина стебля.

Методы класса:

* вывод информации о цветке (с возвратом значения),
* подсчет стоимости покупки букета (количество цветов в букете задается в виде параметра, void-метод).
* дополнительно придумайте к каждому классу еще по одному методу, возвращающему значение.

**Вариант 3.** Класс «Спортсмен»

Поля класса: имя, отчество, фамилия, пол, возраст (\*дата рождения), рост, вес, вид спорта.

Методы класса:

* вывод информации о спортсмене;
* определение идеального веса по формуле Брока: до 40 лет: *идеальный вес (кг) = рост (см) – 110* после 40 лет: *идеальный вес (кг) = рост (см) – 100*  определение идеального веса по формуле Купера:

для женщины: *идеальный вес (кг) = (рост (см) х 3,5 : 2,54 - 108) х 0,453* для мужчины: *идеальный вес* *(кг) = (рост (см) х 4,0 : 2,54 - 128) х 0,453.*

* Дополнительно придумайте к каждому классу еще по одному методу, возвращающему значение.

**Вариант 4.** Класс «Комната»

Поля класса: длина комнаты, ширина комнаты, высота потолков, количество окон, высота окна, ширина окна.

Методы класса:

* «Вычисление площади комнаты»;
* «Вычисление объема комнаты»;
* «Информация о комнате»;
* «Ремонт»: - определить количество рулонов обоев, необходимое для ремонта данной комнаты. Предусмотреть работу с рулонами разной длины (10 м и 15 м).
* дополнительно придумайте к каждому классу еще по одному методу, возвращающему значение.

**Вариант 5.** Класс «Автомобиль».

Поля класса: объем бака (начальное значение этого поля установите прямо в классе), количество бензина в баке (начальное значение - ноль), расход бензина на сотню (начальное значение установите в классе), пробег.

Методы класса:

* «Информация о машине» - отображаются следующие характеристики:

объем бака, количество бензина в баке, расход и пробег.

* «Заправить машину» - позволяет увеличивать количество бензина в баке. Если весь желаемый бензин поместился, то метод должен возвращать ноль. Если что-то не поместилось, то метод должен возвращать это значение.
* «Ехать» - в качестве параметра будет количество километров. Во время поездки бензин в баке должен уменьшаться, пробег увеличиваться. При этом необходимо отследить ситуацию, когда бензина может не хватить на всю поездку. В этом случае нужно выдать сообщение, что столько-то километров не доехали. В случае, когда пробег пройдет порог очередной тысячи - расход надо увеличить на 0,1 и вывести сообщение.
* «Надо ли заправляться?» - метод возвращает значение «Истина», если в баке осталось меньше 10% бензина и «Ложь» – в противном случае.

**Вариант 6.** Класс «Покупатель»

Поля класса: название покупаемого продукта, цена, количество, кошелек, настроение (начальное значение=10).

Методы класса:

* «Информация о покупке» - отображается название купленного продукта, его количество и стоимость покупки.
* «Купить» - если человек что-то покупает, то стоимость покупки увеличивается, а наличность кошелька уменьшается. Выдавать сообщение, сколько денег осталось в кошельке после совершения покупки. Добавьте связь с настроением: если денег хватило на совершение покупки, то настроение увеличивается на 50% от суммы покупки, а если не хватило – то настроение уменьшается на количество единиц, равное количеству денег, которых не хватило для совершения покупки.
* «Посмотреть настроение» - отслеживайте само настроение:

если настроение больше 15, то оно «бодрое», если от 5 до 15, то – нормальное, если от -10 до 5, то – плохое, если меньше -10, то – депрессивное.

**Критерии оценки:**

**«отлично»**

* правильно решены все задания с 1 по 6;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены все самостоятельные задания №1-2, самостоятельно.

**«хорошо»**

* правильно решены все задания с 1 по 4;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решены все самостоятельные задания №1-2 с недоработками, самостоятельно.

**«удовлетворительно»**

* правильно решены все задания с 1 по 3;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* решено одно самостоятельное задание.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы

**Практическая работа №13 «Создание классов с использованием свойств»**

4 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить создание свойств класса. **Учебные задачи:**

1. Научиться создавать классы с необходимыми свойствами и использовать их в своей программе.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
* 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.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Свойства – неотъемлемая часть класса. Именно с помощью свойств реализовывается один из «китов» объектно-ориентированного программирования – инкапсуляция.

Инкапсуляция (инкапсуляция вариаций)– техника сокрытия частей Объектно-Ориентированных программных систем.

Свойство это конструкция языка C#, которая заменяет собой использование обычных методов доступа.

Работа со свойством экземпляра напоминает работу с полями экземпляра.

Свойство состоит из имени, типа и тела. В теле задаются методы доступа, через использование ключевых слов **set** и **get**.

Метод **set** автоматически срабатывает тогда, когда свойству пытаются присвоить значение. Это значение представлено ключевым словом **value**.

Метод **get** автоматически срабатывает тогда, когда мы пытаемся получить значение.

**int** field;

public int Property

{ **get** { **return** field; } **set** { field = **value**;} }

Метод доступа **get**–используется для получения значения из переменной. Метод доступа **set**-используется для записи значения в переменную.

|  |  |
| --- | --- |
| Свойство только для чтения int field;  public int Property  { get {  return field;  }  } | Свойство только для записи  int field;  public int Property  { set {  field = value;  }  } |

Автоматически реализуемые свойства это более лаконичная форма свойств, их есть смысл использовать, когда в методах доступа **get** и **set** не требуется дополнительная логика.

При создании автоматически реализуемых свойств, компилятор создаст закрытое, анонимное резервное поле, которое будет доступно с помощью методов **get** и **set** свойства.

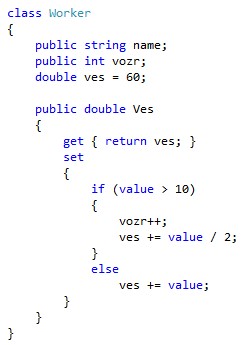
**Вопросы для закрепления теоретического материала:**

1. Для чего нужны свойства класса?
2. Что означает «инкапсуляция»?
3. Опишите механизм создания свойства класса.

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Измените программу из практической работы №12 задания №1, так чтобы вместо двух методов **SetEat** и **GetEat** появилось одно свойство **Ves**.

* Измените класс **Worker** в соответствии с рисунком 133. Все имена полей измените с маленькой буквы.



Поля в классе пишутся с прописной

буквы

Имя свойств класса пишется с заглавной буквы. Дан-

ное свойство предназначено для чтения и записи пе-

ременной.

Рисунок 133 – Код изменения класса - Измените событие на кнопку в соответствии с рисунком 134.



Рисунок 134 – Код события нажатия на кнопку

* Запустите программу на выполнение. Проверьте работоспособность.

**Самостоятельное задание № 1.** Вернитесь к одному из своих ранее составленных проектов, работающих с классами, и измените классы так, чтобы в них появились свойства.

**Критерии оценки:**

**«отлично»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, самостоятельно.  **«хорошо»**
* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, не все методы чтения и записи закрытых полей заменены на свойства.

**«удовлетворительно»**

* правильно решено задание 1;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* не решено самостоятельное задание №1.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы

**Практическая работа №14 «Создание классов, иерархически связанных между собой»**

2 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить создание классов, связанных отношением наследования.
3. Освоить написание конструкторов.
4. Освоить переопределение методов. **Учебные задачи:**

1. Научиться создавать и использовать иерархически связанные между собой классы в своих проектах.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
* 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.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Наследование – одна из основных характеристик объектно-ориентированного программирования. Когда новый класса становится потомком существующего, он наследует все незакрытые методы, свойства и поля своего предка. Новый класс затем можно расширять, добавляя или заменяя эти методы, поля и свойства. Все это, в конечном счете, позволяет значительно упростить программирование путем многократного использования кода.

Синтаксис:

class имя\_класса : имя\_родительского\_класса {тело\_класса}

Конструктор класса—специальный метод, который вызывается вовремя построения класса (Рисунок 135).

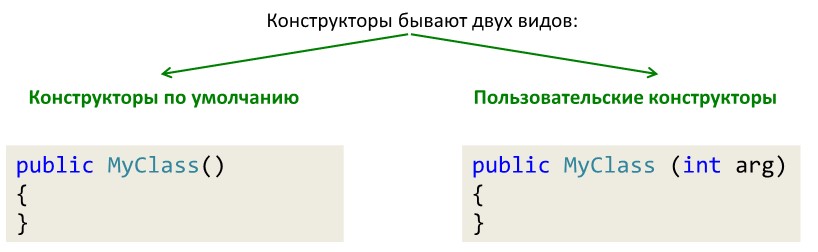


Рисунок 135 – Виды конструкторов

Если в теле класса не определен явно ни один конструктор, то всегда используется «невидимый» конструктор по умолчанию.

Имя конструктора всегда совпадает с именем класса. Конструкторы не имеют возвращаемых значений.

Задача конструктора по умолчанию–инициализация полей значениями по умолчанию.

Задача пользовательского конструктора–инициализация полей предопределенными пользователем значениями.

**Вопросы для закрепления теоретического материала:**

1. Как вы понимаете суть наследования?
2. Какие методы есть у любого класса?
3. Каким ключевым словом необходимо снабдить метод предка, чтобы его потомки могли этот метод изменять?
4. Какие типы конструкторов вы знаете?
5. Зачем использовать конструкторы по умолчанию?

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Создать классы «фигура» и «треугольник» так, чтобы «фигура» был «предком», а «треугольник» - потомком.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form. - Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Работа с классом фигура**.
* Разместите компоненты **Button, TextBox, Label** как показано на рисунке. В свойстве **Font** установите размер шрифта равным **12** пт. В свойствах **Text** задайте значения, соответствующие рисунку 136.

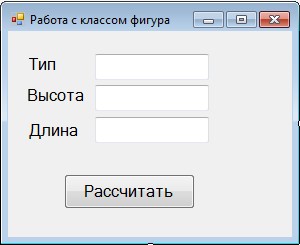


Рисунок 136 – Интерфейс приложения

* В обозревателе решений нажмите ПКМ и в контекстном меню выберите команду **Добавить** – **класс** (Рисунок 137).

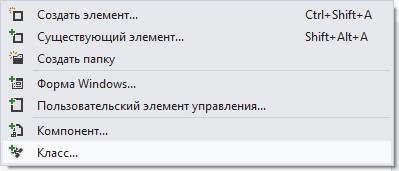
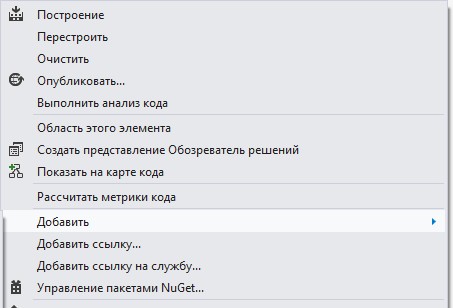


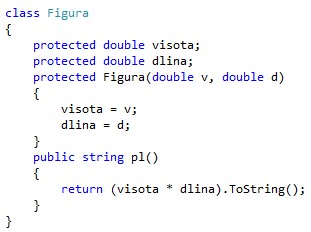
Рисунок 137 – Добавление нового класса в проект

* В окне добавления нового элемента задайте имя класса **Figura** и нажмите кнопку **Добавить**.
* В результате в код проекта добавиться заготовка для создания класса **Figura** (Рисунок 138).



Рисунок 138- Заготовка для создания класса

* Напишите класс, включив в него методы, поля и конструктор в соответствии с рисунком 139.



Конструктор, который задает началь-

ные свойства данных полей.

Общедоступный

метод,

который возвращает значение

площади

фигуры

строчного

Общедоступный метод, который

воз-

вращает значение площади фигуры

строчного типа.

Защищенные поля, но доступные для наследуе-

мых классов:

**высота**

и

**длина**

. Данные свой-

ства есть у всех фигур

Рисунок 139 – Код класса

* В обозревателе решений нажмите ПКМ и в контекстном меню выберите команду **Добавить** – **класс** (Рисунок 140).

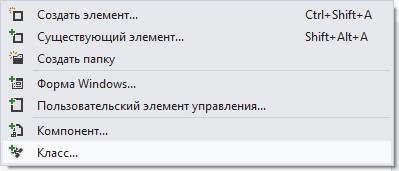
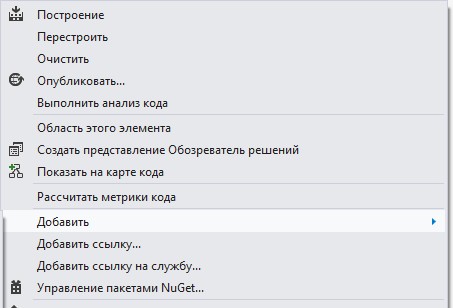
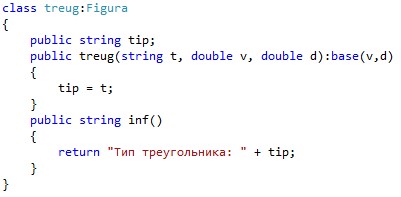


Рисунок 140 – Добавление нового класса в проект

* В окне добавления нового элемента задайте имя класса **Treug** и нажмите кнопку **Добавить**.
* В результате в код проекта добавиться заготовка для создания класса **Treug.**
* Напишите класс, включив в него методы, поля и конструктор в соответствии с рисунком 141.



Указывается, что

данный класс наследует поля и методы класса

Конструктор указывает, что значения высоты и длины следует

брать из конструктора предка «Фигура»

Метод, который возвращает строчное значение

типа фигуры

Рисунок 141 – Код класса

* Создайте обработчик событий на кнопку **рассчитать** (Рисунок 142).

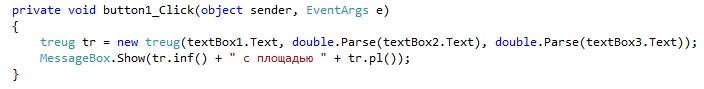


Рисунок 142 – Код события нажатия на кнопку Рассчитать

* Запустите программу на выполнение. Проверьте работоспособность (Рисунок 143).

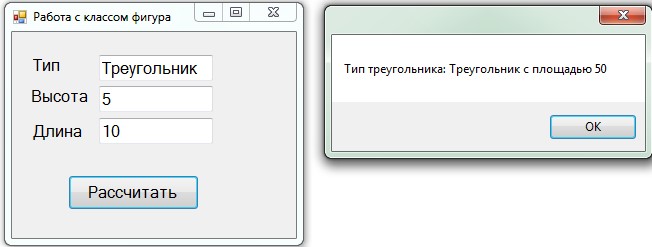


Рисунок 143- Результат тестирования приложения

**Самостоятельное задание № 1.** Построить иерархию классов в соответствии с вариантом задания:

**Вариант 1.** Студент, преподаватель, персона, заведующий кафедрой

**Вариант 2.** Служащий, персона, рабочий, инженер

**Вариант 3.** Рабочий, кадры, инженер, администрация

**Вариант 4.** Деталь, механизм, изделие, узел

**Вариант 5.** Организация, страховая компания, нефтегазовая компания, завод

**Вариант 6.** Журнал, книга, печатное издание, учебник

**Вариант 7.** Тест, экзамен, выпускной экзамен, испытание

**Вариант 8.** Место, область, город, мегаполис

**Вариант 9.** Игрушка, продукт, товар, молочный продукт

**Вариант 10.** Квитанция, накладная, документ, счет

**Вариант 11.** Автомобиль, поезд, транспортное средство, экспресс

**Вариант 12.** Двигатель, двигатель внутреннего сгорания, дизель, реактивный двигатель

**Вариант 13.** Республика, монархия, королевство, государство

**Вариант 14.** Млекопитающее, парнокопытное, птица, животное

**Вариант 15.** Корабль, пароход, парусник, корвет

**Критерии оценки:**

**«отлично»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, самостоятельно.

**«хорошо»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, не все поля, методы и конструкторы реализованы, а также отсутствует наследование классов.

**«удовлетворительно»**

* правильно решено задание 1;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* не решено самостоятельное задание №1.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы

**Практическая работа №15 «Создание проектов с использованием текстовых файлов»**

2 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить приемы работы с текстовыми файлами. **Учебные задачи:**

1. Научиться создавать проекты для работы с текстовыми файлами.

**Образовательные результаты, заявленные во ФГОС ТОП-50:**

Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.
4. Оформить отчет.

**Краткие теоретические материалы по теме практической работы**

Файл — это именованная структура данных, представляющая собой последовательность элементов данных, причем количество элементов последовательности практически не ограничено. В первом приближении файл можно рассматривать как массив переменной длины неограниченного размера.

В пространстве имен **System.IO** предусмотрено четыре класса, которые предназначены для работы с файловой системой компьютера, т. е для создания, удаления переноса и т.д. файлов и каталогов:

* **Directory** и **File** реализуют свои возможности с помощью статических методов, поэтому данные классы можно использовать без создания соответствующих объектов (экземпляров классов);
* **DirectoryInfo** и **FileInfo** обладают схожими функциональными возможностями c **Directory** и **File**, но порождены от класса **FileSystemInfo** и поэтому реализуются путем создания соответствующих экземпляров классов.
* Для записи и чтения текстовых файлов очень удобно использовать классы **StreamWriter** и **StreamReader**.

Для записи в файл нужно сначала создать файл и открыть поток для работы с ним

StreamWriter sw = File.CreateText("test.txt"); sw - поток

File.CreateText("test.txt") - метод создания файла «test.txt»(по умолчанию сохраняется в папку Debug сохраненного проекта).

После того как открыт поток можо записывать в него текстовые строки используя методы **Write** и **WriteLine**.

sw.WriteLine("Мой первый файл созданый в C#");

Если проверить после запуска программы файл, то увидим что он пуст.

Данные не появятся в файле до тех пор пока мы не закроем поток sw. Делается это с помощью метода Close: sw.Close();

Добавить текст в уже созданный файл понадобится метод **AppendText**. Использовать его нужно вместо **CreateText**:

StreamWriter sw = File.AppendText("test.txt"); sw.WriteLine("Добавили вторую строчку в файл"); sw.Close();

Для чтения нужно открыть поток класса **StreamReader**, привязав его к файлу. Делается это при помощи метода

File.OpenText(‘имя файла’);

StreamReader sr = File.OpenText("test.txt"); Если нужно считать только первую строку из файла для этого воспользуемся следующей конструкцией **sr.ReadLine().**

Допустим нужно вывести первую строчку из файла на консоль. Делается следующим образом:

System.Console.WriteLine(sr.ReadLine());

Чтение из файла проводится построчно. Для вывода всего файла нам нужно построчно его считывать до тех пор, пока в нем есть строчки.

while (true)

{

string st = sr.ReadLine(); if (st==null) break;

System.Console.WriteLine(st);

}

Как и при записи в файл, так и при чтении из него не стоит забывать о закрытии потока методом **Close**.

Sr.Close();

**Вопросы для закрепления теоретического материала:**

1. Что такое файл? В чем заключаются особенности текстового файла?
2. В чем преимущество использования файлов по сравнению с массивами?
3. Какие операции можно производить с файлами?
4. Может ли файл состоять из одного элемента?
5. Какой файл называется текстовым?
6. Можно ли текстовый файл открыть одновременно для чтения и для записи?

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Создайте приложение, которое читает и выводит информацию в текстовый файл.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form. - Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Работа с файлами**.
* Разместите на форме 5 кнопок и текстовое поле и в текстовом поле свойство **Multiline** = **true** (включение многострочного режима) как показано на рисунке 144. В свойстве **Font** установите размер шрифта равным **12** пт.

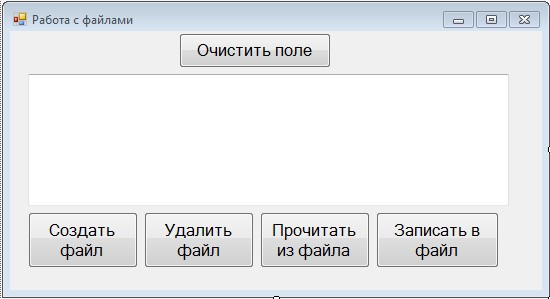


Рисунок 144 – Интерфейс приложения

* Создайте событие для элемента **Button1(Создать файл)**, для этого достаточно дважды щелкнуть мышью по данному элементу. Выше метода **Button1\_Click** подключите библиотеку System.IO, для работы с файлами (Рисунок 145).

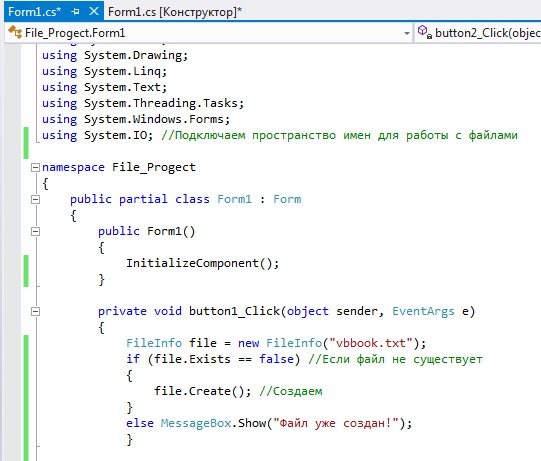


Рисунок 145 – Код события нажатия на кнопку Создать файл

* Создайте событие для элемента **Button2(Удалить файл)**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 146).

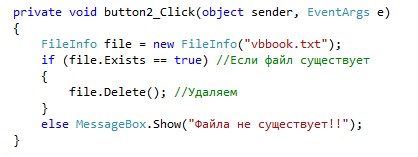


Рисунок 146 – Код события нажатия на кнопку Удалить файл

* Создайте событие для элемента **Button3(Прочитать из файла)**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 147).

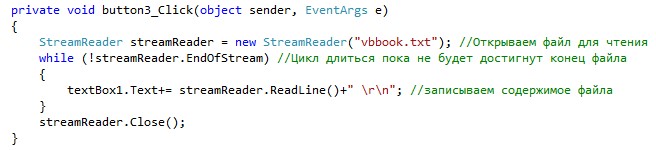


Рисунок 147 – Код события нажатия на кнопку Прочитать из файла

* Создайте событие для элемента **Button4(Записать в файл)**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 148).

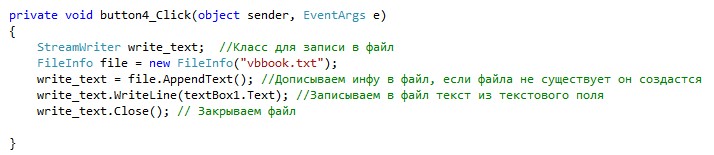


Рисунок 148 – Код события нажатия на кнопку Записать в файл

* Создайте событие для элемента **Button5(Очистить поле)**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 149).

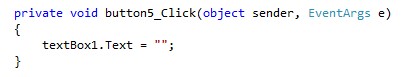


Рисунок 149 – Код события нажатия на кнопку Очистить поле

* Запустите проект и протестируйте работоспособность (Рисунок 150). Запишите в файл числовые значения.

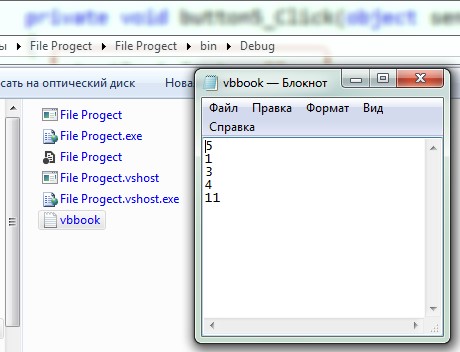
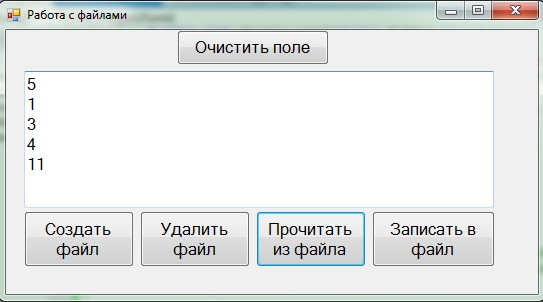


Рисунок 150- Результат тестирования приложения

* Добавьте на форму кнопку, по нажатию на которую будет происходить подсчет суммы цифр из файла. Установите свойства для компонентов в соответствии с рисунком 151. Свойство Text компонент **Label1** оставьте пустым.

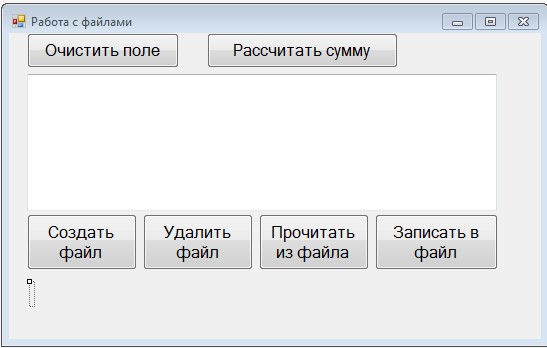


Рисунок 151 – Интерфейс приложения

* Создайте событие для элемента **Button6(Рассчитать сумму)**, для этого достаточно дважды щелкнуть мышью по данному элементу (Рисунок 152).



Рисунок 152 – Код события нажатия на кнопку Рассчитать сумму - Запустите проект и протестируйте работоспособность (Рисунок 153).

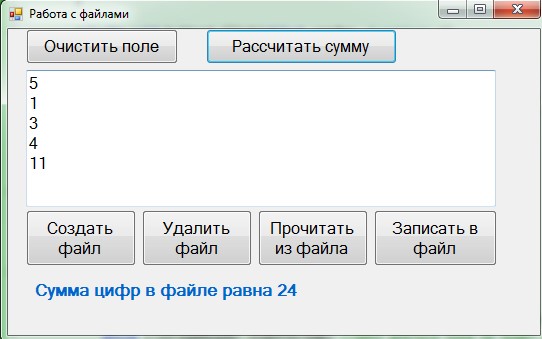


Рисунок 153- Результат тестирования приложения

**Самостоятельное задание №1.** Модифицируйте проект задания №1 по вариантам. Вывод результата поместить в конец файла, пример изображен на рисунке 154.

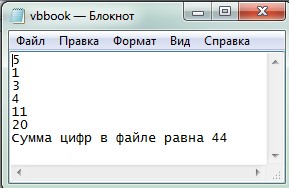
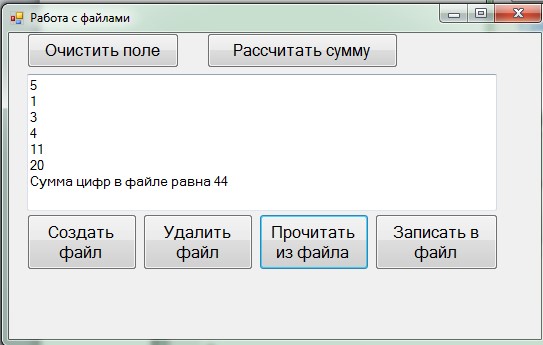


Рисунок 154- Результат тестирования приложения **Вариант 1.** количество чисел в файле.

**Вариант 2.** максимальный элемент в файле.

**Вариант 3.** среднее арифметическое значение чисел в файле.

**Вариант 4.** минимальный элемент в файле.

**Вариант 5.** количество отрицательных чисел в файле.

**Вариант 6.** количество положительных чисел в файле.

**Вариант 7.** количество нулей в файле.

**Вариант 8.** произведение чисел в файле.

**Вариант 9.** количество искомого числа в файле.

**Вариант 10.** количество максимальных элементов в файле.

**Вариант 11.** количество минимальных элементов в файле.

**Вариант 12.** количество четных чисел в файле.

**Вариант 13.** количество не четных чисел в файле.

**Вариант 14.** количество чисел, которые делятся на 3 без остатка.

**Вариант 15.** количество чисел, которые делятся на 7 без остатка. **Вариант 16.** сумму двузначных чисел.

**Критерии оценки:**

**«отлично»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, самостоятельно и вариант соответствует номеру ПК.

**«хорошо»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание 1с недочетами, вывод осуществляется не в файл, а на форму и вариант соответствует номеру ПК.

**«удовлетворительно»**

* правильно решено задание 1;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* решено самостоятельное задание, но вариант не соответствует номеру ПК и вывод результаты осуществляется на форму.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы

**Практическая работа №16 «Создание MDI приложений и меню»**

2 часа

**Учебная цель:**

1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
2. Освоить приемы работы с окнами и меню. **Учебные задачи:**

1. Научиться создавать проекты, которые содержат несколько окон.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.

**Краткие теоретические материалы по теме практической работы**

Создание приложения с дочерними окнами в **Visual** **Studio** предельно просто. При создании **Windows** приложения достаточно установить свойство **IsMdiContainer** в значение **true** и проект решения становится проектом решения с дочерними окнами, приобретая соответствующий вид и свойства.

**Вопросы для закрепления теоретического материала:**

1 Какое свойство следует задать для того, чтобы форму сделать дочерней?

**Задания и инструкция по выполнению практической работы Задание 1.** Создайте многооконное приложение.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Многооконное приложение**.
* Установите свойство **IsMdiContainer** для формы в значение **True** (Рисунок 155).



Рисунок 155 – Значение свойства IsMdiContainer - Разместите компонент **MenuStrip** (Рисунок 156).

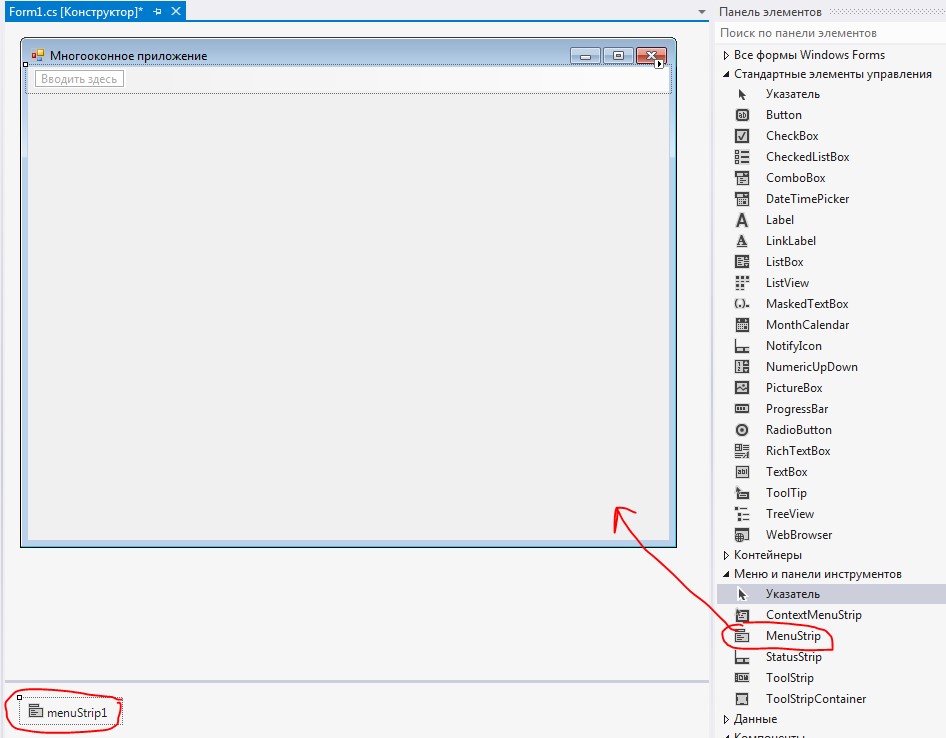


Рисунок 156 – Размещение невизуального компонента на форме - Добавьте пункты меню как показано на рисунке 157.

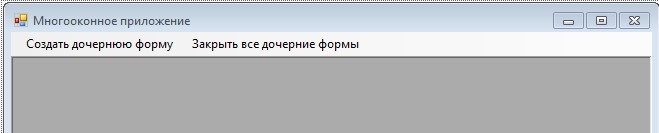


Рисунок 157 – Интерфейс приложения

-Добавьте к проекту новую форму, для этого следует нажать правой кнопкой мыши в **Обозревателе решений** на проекте и в контекстном меню выбрать пункт **Добавить**, затем вызвать команду **Создать элемент...** (Рисунок

158)

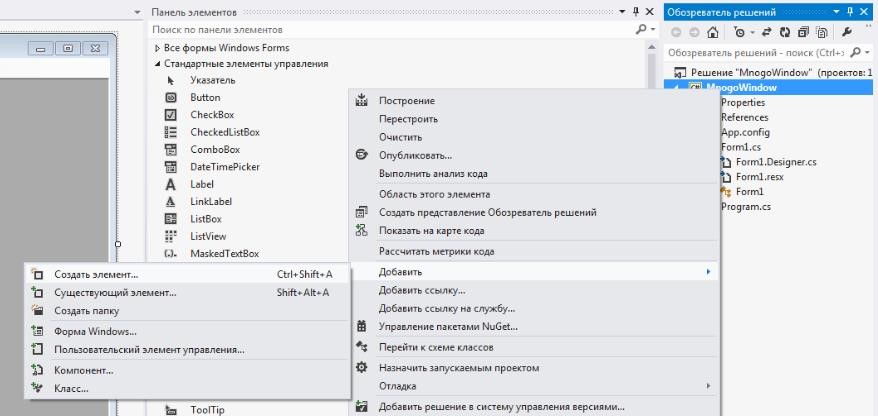


Рисунок 158 – Добавление новой формы к проекту

* В окне **Добавление** **нового** **элемента** выберите элемент **Форма** **Windows** **Forms** и нажмите на кнопку **Добавить** (Рисунок 159).

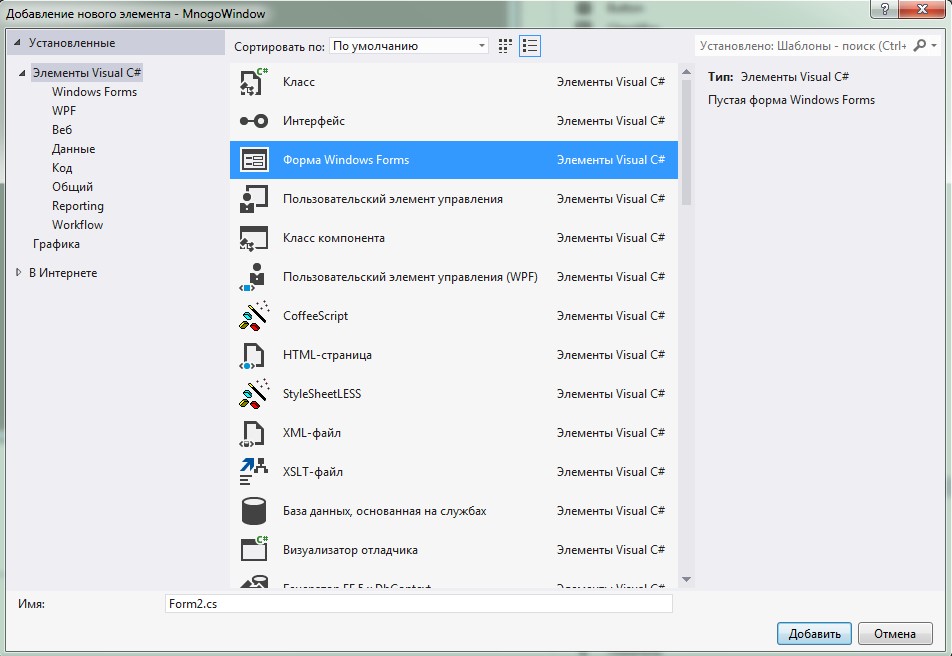


Рисунок 159 – Диалоговое окно добавления нового элемента

* Создайте обработчик события пункта меню **Создать дочернюю форму**, для этого следует щелкнуть мышью на данном пункте меню двойным щелчком (Рисунок 160).

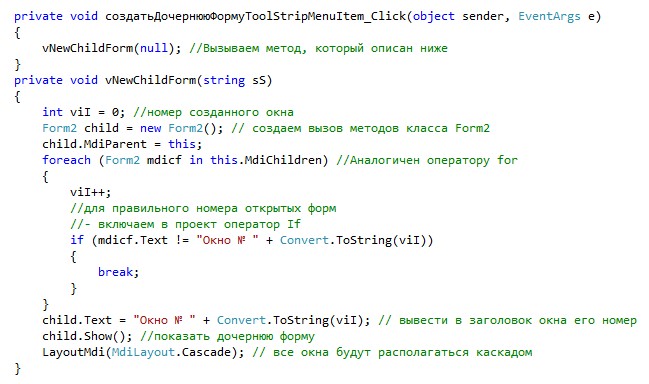


Рисунок 160 – Код события нажатия на пункт меню Создать дочернюю форму

* Создайте обработчик события пункта меню **Закрыть все дочерние формы**, для этого следует щелкнуть мышью на данном пункте меню двойным щелчком (Рисунок 161).

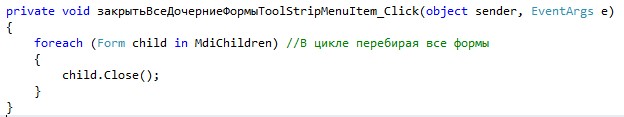


Рисунок 161 – Код события нажатия на пункт меню Закрыть все дочерние формы

* Запустите проект и протестируйте работоспособность (Рисунок 162).

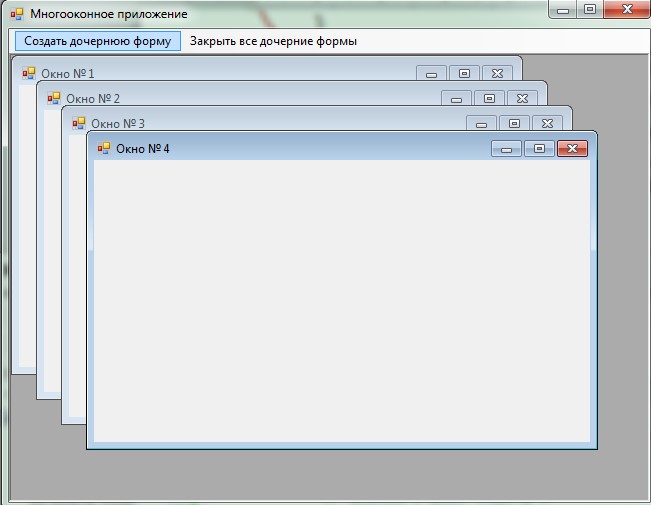


Рисунок 162 - Результат тестирования приложения

**Задание 2.** Добавьте возможность передачи сообщений между формами. - Добавьте новый пункт меню **Прием сообщений** (Рисунок 163).

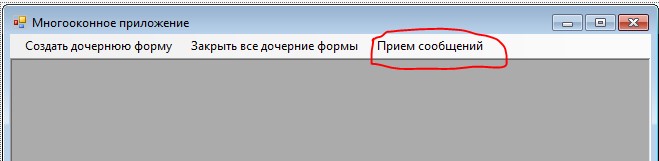


Рисунок 163 – Добавление нового пункта меню

* Создайте обработчик события данного пункта меню, для этого следует щелкнуть мышью на данном пункте меню двойным щелчком. Метод **vSetPhrase** подчеркнут, как ошибка, т. к. он пока не написан (Рисунок 164).

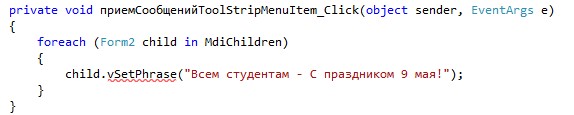


Рисунок 164 – Код события пункта меню Прием сообщений

* Создайте функцию для приема сообщения от дочерних окон (Рисунок 165).

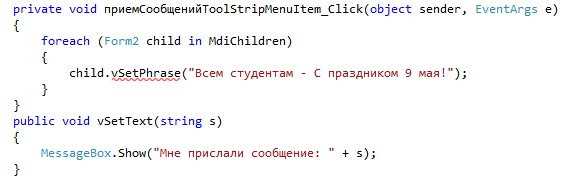


Рисунок 165- Функция для приема сообщения от дочерних окон

* На дочернюю форму, поместите **Button** и **Label**, с соответствующим текстом, как показано на рисунке 166.

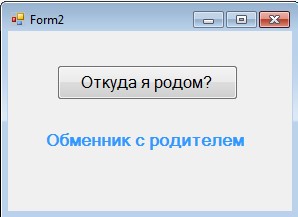


Рисунок 166 – Интерфейс формы 2

* В коде для дочерней формы добавим еще один конструктор, который позволяет принимать ссылку на основную форму (Рисунок 167).

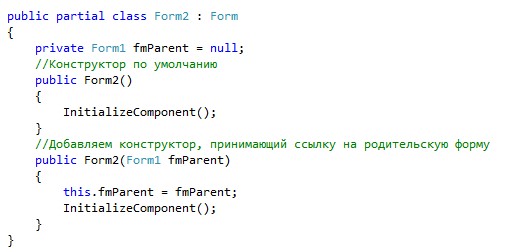


Рисунок 167 – Код формы 2

* В основной форме **Form1** в функции создания дочернего окна необходимо заменить (Рисунок 168).

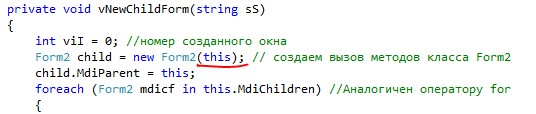


Рисунок 168 – Код формы 1

* В код **Form2** допишите функцию приема данных от основной формы, и отсылки сообщений из дочерней формы в основную (Рисунок 169).

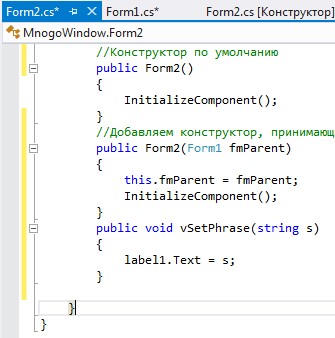


Рисунок 169 – Код формы 2

* Для **Form2** создайте событие **FormClosed**, для этого выделить **Form2** в инспекторе объектов переключитесь на методы и напротив данного метода щелкните двойным щелчком мыши (Рисунок 170).

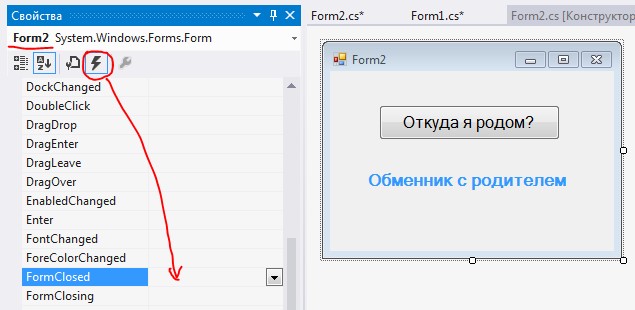


Рисунок 170 – Создание события -Напишите строку кода передачи сообщения (Рисунок 171).

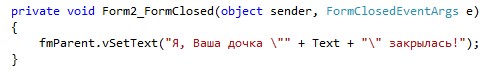


Рисунок 171 – Код события закрытия формы 2

* Создайте обработчик события для кнопки **Откуда я родом?**, помещенной на **Form2**, для этого следует щелкнуть мышью двойным щелчком (Рисунок 172).

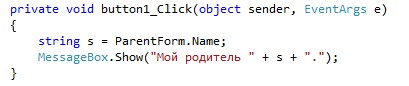


Рисунок 172 – Код события нажатия на кнопку Откуда я родом?

* Запустите проект и протестируйте работоспособность (Рисунок 173).

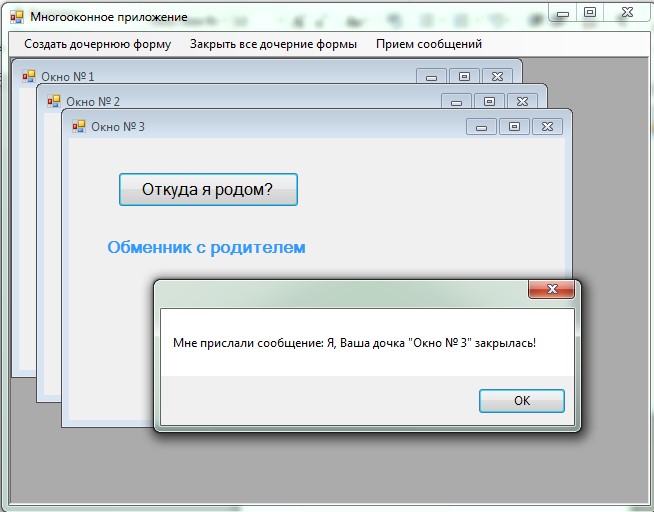
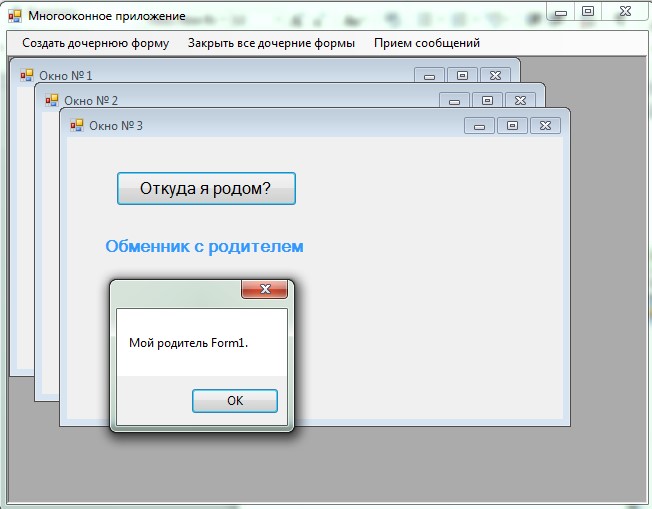
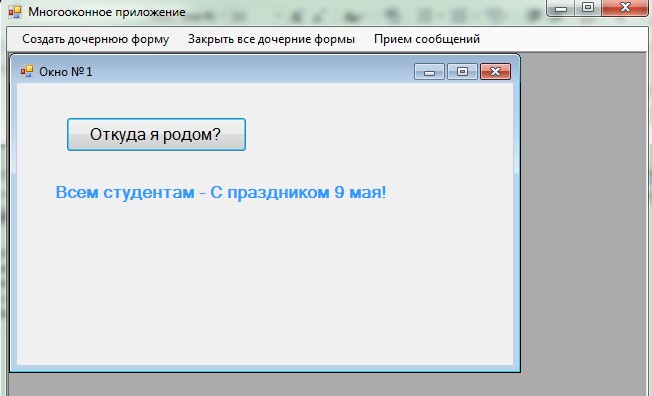


Рисунок 173- Результат тестирования приложения

**Критерии оценки:**

**«отлично»**

* правильно решено задание 1 и 2.

**«хорошо»**

* правильно решено задание 1 и 2 с помощью преподавателя. **«удовлетворительно»**  правильно решено задание 1.

**Практическая работа №17 «Использование компонентов OpenFileDialog и**

**SaveFileDialog для работы с файлами»**

2 часа

**Учебная цель:**

* 1. Познакомиться с интегрированной средой разработки Microsoft Visual Studio.
  2. Освоить приемы работы с компонентами OpenFileDialog и

SaveFileDialog. **Учебные задачи:**

1. Научиться создавать проекты с использование компонентов

OpenFileDialog и SaveFileDialog.

**Образовательные результаты, заявленные во ФГОС ТОП-50:** Студент должен уметь:

* 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. знать:
* Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

**Задачи практической работы:**

1. Повторить теоретический материал по теме практической работы.
2. Ответить на вопросы для закрепления теоретического материала.
3. Выполнить задания.

**Краткие теоретические материалы по теме практической работы**

OpenFileDialog элемент управления используется, чтобы просмотреть и выбрать файл на компьютере.

Свойство InitialDirectory представляет каталог, в который будет отображаться при открытии окна выбора файла появляется впервые.

openFileDialog1.InitialDirectory = @"С:\";

Если RestoreDirectory свойство установлено в True, что при открытии файла восстанавливается текущий каталог до закрытия.

openFileDialog1.RestoreDirectory = истина;

Свойство title используется для задания или получить звание диалог открытия файла.

openFileDialog1.Название = "Просмотреть Текстовые Файлы";

Свойство Filter представляет собой фильтр в диалоге открытия файла, который используется, чтобы выбирать тип файлов, которые будут загружены во время просмотра опция в диалоге открытия файла. Например, если вам нужно ограничить пользователей в изображениях, мы можем установить свойство **Filter**, чтобы загрузить файлы изображений только.

openFileDialog1.Фильтр = "текстовые файлы (txt)(\*.txt)|\*.txt|все

файлы(\*.\*)|\*.\*";

Свойство FilterIndex представляет индекс фильтра, выбранного в диалоговом окне файла.

openFileDialog1.FilterIndex = 2;

Свойство **MultiSelect** - при значении **true** позволяет выбрать мышкой при нажатой кнопке **Shift** или **Ctrl** несколько файлов и сохранить их имена в свойстве **FileNames** в виде массива строк.

Свойство **ReadOnlyChecked** - при значении **true** позволяет открывать команде **OpenFile** выбранные файлы только в режиме чтения.

Все рассмотренные свойства так же можно применить к компоненту **SaveFileDialog.**

**Вопросы для закрепления теоретического материала:**

1. Какое свойство отвечает за фильтрацию файлов в окне?
2. Какое свойство позволяет поменять заголовок окна выбора файла?
3. Какое свойство позволяет установить директорию по умолчанию?

**Задания и инструкция по выполнению практической работы**

**Задание 1.** Создайте приложение, позволяющее открывать и сохранять файлы.

* Откройте Visual Studio, далее нажмите на Начальной странице пункт меню Создать проект и в разделе Visual C# выберите Приложение Windows Form.
* Задайте имя проекта и нажмите **ОК**.
* Измените заголовок формы, в свойстве **Text** установите значение **Работа с файлами**.
* Поместите на форму компоненты **RichTextBox**, **Button**, **OpenFileDialog** и **SaveFileDialog**. Компоненты **OpenFileDialog** и **SaveFileDialog** не визуальные и, потому, разместились ниже формы (Рисунок 174).

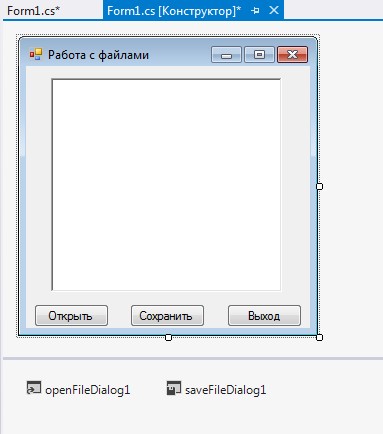


Рисунок 174 – Интерфейс приложения

* Дважды кликнем на форме левой кнопкой мыши, таким образом, создадим обработчик **Form1\_Load**, который будет вызываться при каждой загрузки формы и который в дальнейшем будет часто использовать для начальной инициализации параметров программы и переменных. В обработчике запишем код очистки **RichTextBox** (Рисунок 175):

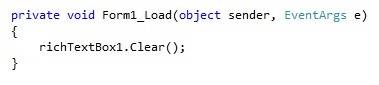


Рисунок 175 – Код события при открытиии формы - Создайте обработчик события для кнопки **Выход** (Рисунок 176).

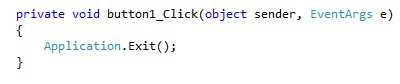


Рисунок 176 – Код события нажатия на кнопку Выход

* Установим для **OpenFileDialog1** следующие значения свойств (Рисунок 177):

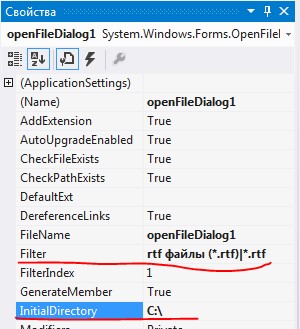


Рисунок 177 – Установка значения свойств - Подключим библиотеку для работы с файлами **IO** (Рисунок 178).

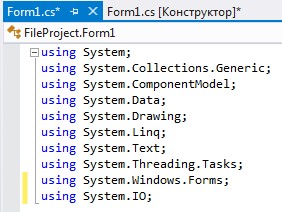


Рисунок 178 – Подключение библиотеки

* Создайте обработчик события для кнопки **Открыть** (Рисунок 179).

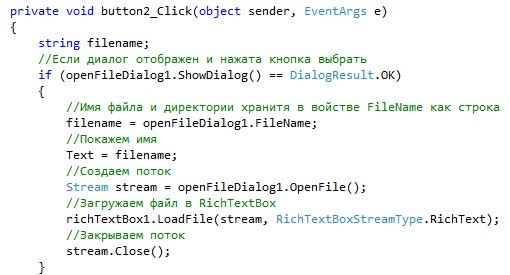


Рисунок 179 – Код события нажатия на кнопку Открыть

* Установите для **SaveFileDialog**1 следующие значения свойств (Рисунок 180):

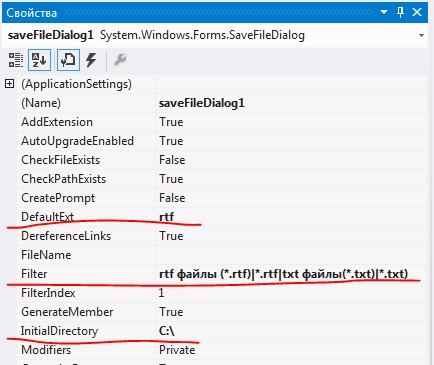


Рисунок 180 – Установка значения свойств

* Для обработчика события нажатия кнопки **Сохранить** напишите следующий код (Рисунок 181):

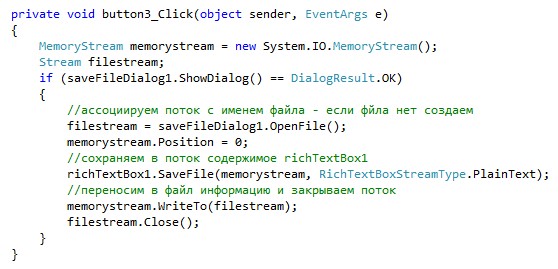


Рисунок 181 – Код события нажатия на кнопку Сохранить

* Запустите проект и протестируйте работоспособность. Создайте с помощью текстового редактора документ с расширением **rtf**, загрузить его с помощью кнопки **Открыть**. протестируйте кнопку **Сохранить**, набрав новый текст или изменив загруженный (Рисунок 182).



Рисунок 182- Результат тестирования приложения

**Самостоятельное задание №1.** Выполните вариант заданий, указанный преподавателем.

**Вариант 1.** Создайте приложение, позволяющее написать электронное письмо и сохранить его на диске.

**Вариант 2.** Создайте приложение позволяющее проводить анкетирование по заранее подготовленным в текстовом файле вопросам и сохранять результаты на внешнем носителе.

**Вариант 3.** Создайте приложение позволяющее принимать заявки от пользователей на ремонт компьютерной техники и сохранять их на внешнем носителе.

**Критерии оценки:**

**«отлично»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, самостоятельно.

**«хорошо»**

* правильно решено задание 1;
* наличие отчета по практической работе, соответствующего образцу оформления отчета;
* решено самостоятельное задание №1, с ошибками.

**«удовлетворительно»**

* правильно решено задание 1;
* наличие отчета по практической работе, не соответствующего образцу оформления отчета.
* не решено самостоятельное задание №1.

**Отчет по практической работе** аналогичный практической работе №4, в отчет входит оформление всех самостоятельных заданий данной практической работы.