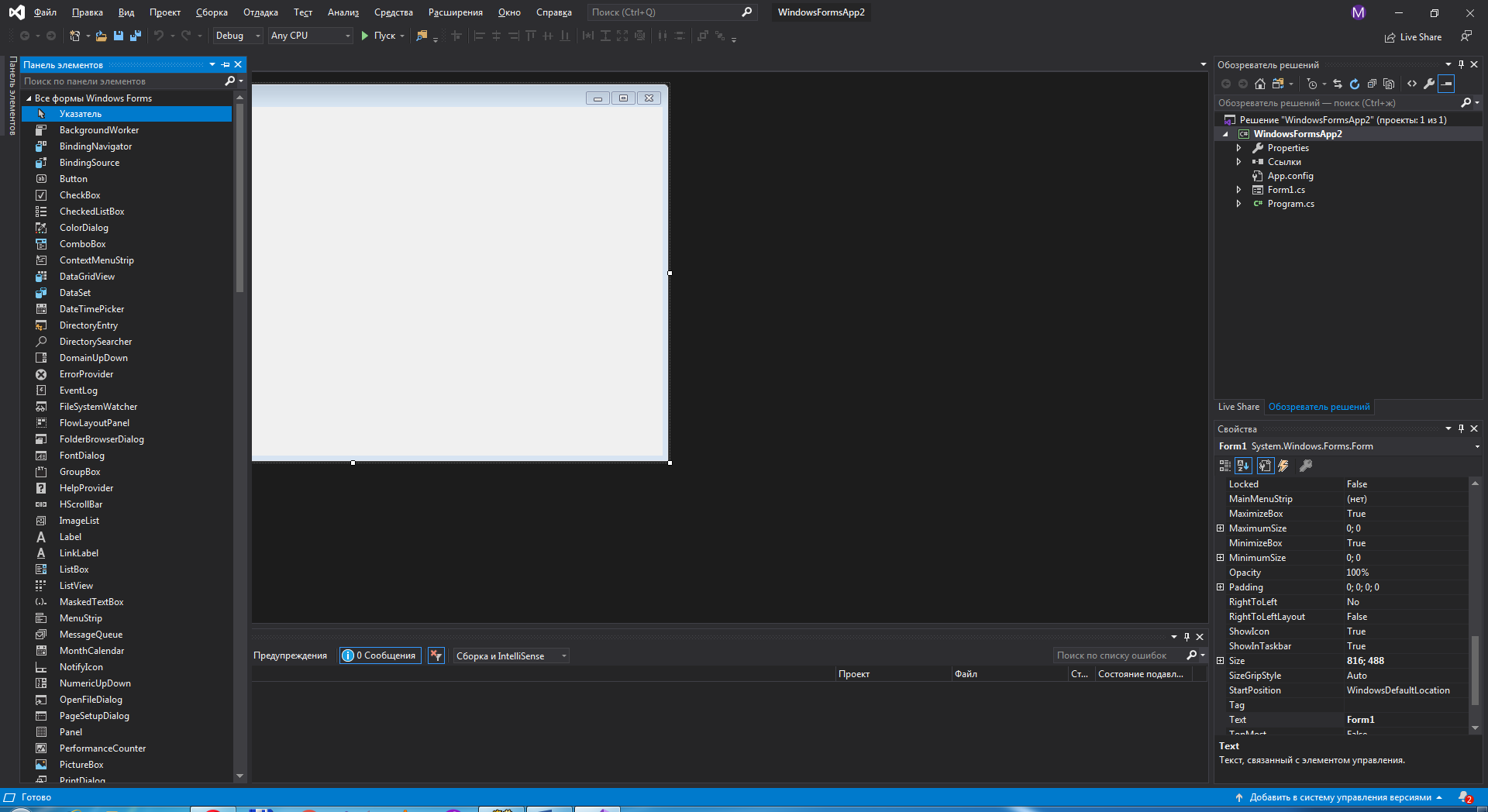
Знакомство с Visual Studio и язык программирования C#.

**Создание проекта.**

1. Запустить Visual Studio.
2. В открывшемся окне выбираем «Создание проекта»
3. Выбираем из списка => Язык C# => «Приложение Windows Form (.Net Framework)»
4. Задаем имя проекта. Расположение.

У вас откроется рабочее окно.



1. Нажмите на Панель элементов и зафиксируйте его для удобства (Элементы Form).
2. По центру представлена рабочая область.
3. Обозреватель решений (Отображена иерархия структуры проекта).
4. Свойства (отображает свойства выделенного объекта).
5. Список ошибок (сообщения об ошибках в реальном времени).

Записать в тетрадь этапы создания нового проекта.

Записать в тетради следующие определения элементов:

* Button
* CheckBox
* ComboBox
* Label
* ListBox
* RadioButton
* TextBox
* Timer
* PicturBox
* Chart

**Задание 1. Вычислим сумму, максимального и минимального элемента небольшого множества.**

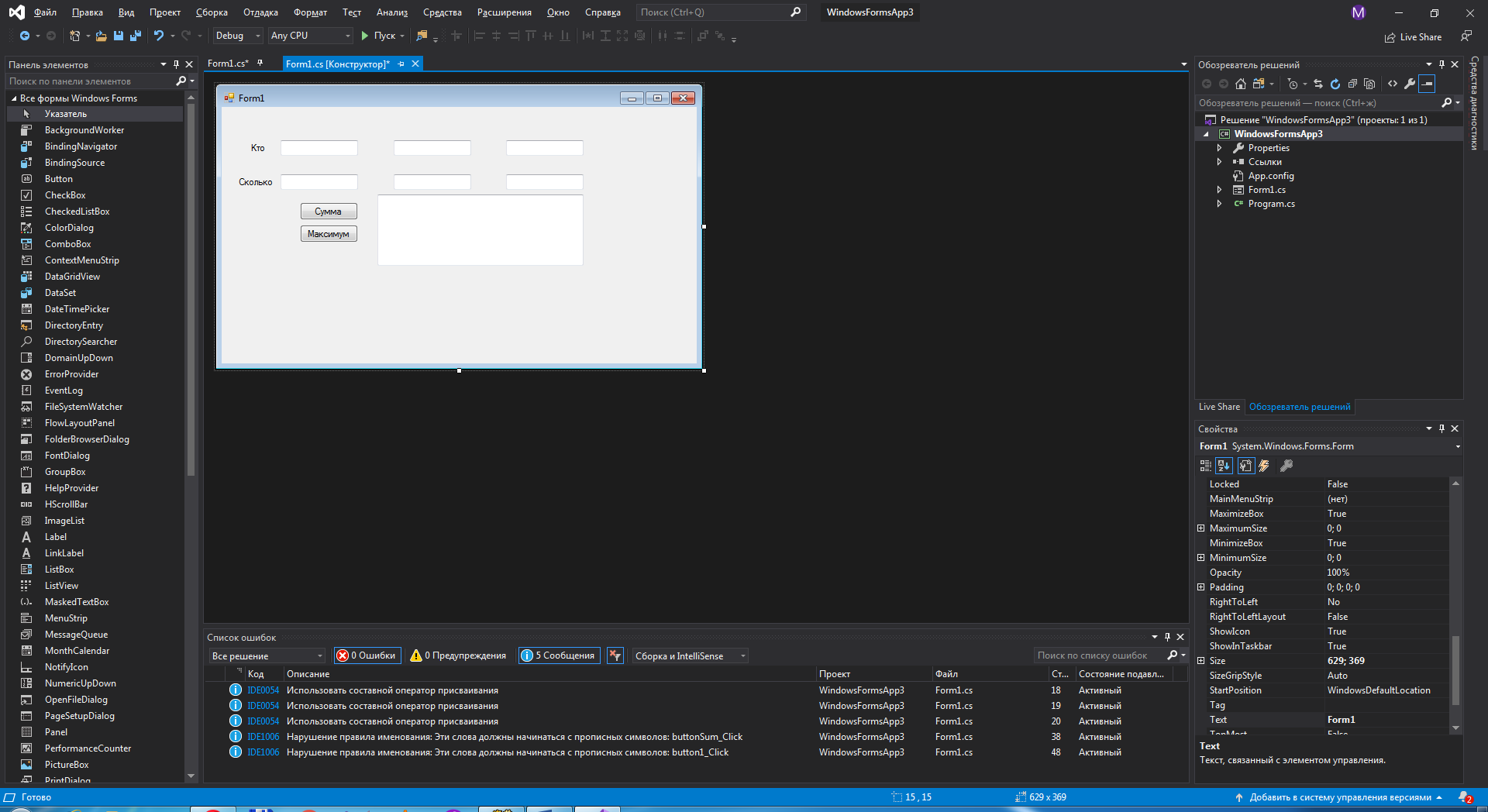
Рассмотрим конкретную задачу:

Аня, Витя и Саша пошли в *лес*. Аня собрала m грибов, Витя – n, Саша – p грибов.

Вот вопросы, которые можно задать в связи с этой задачей:

1. Сколько грибов собрали дети?
2. Какое максимальное число грибов было найдено одним подростком?
3. Кто собрал больше всех грибов?

Ответы на эти вопросы так или иначе требуют вычисления суммы элементов, нахождения максимального (минимального) элемента.



1. На Form1 перенести следующие элементы:

TextBox x7

Label x6

Button x3

Разместите на форме в удобном вам порядке.

Введите текст для Label («Кто», «Сколько»)

Переименуйте Name своих элементов. Например: ButtonSum (кнопка по подсчету суммы)

Подгоните форму по размеру. (Можно сделать захватив за край формы, либо назначит в свойствах Form)

1. Первым делом пропишем функцию для проекта.

Функция - важнейший элемент структурного программирования, позволяющий группировать и обобщать программный код, который может позднее использоваться произвольное число раз.

int sum = 0;//вводим переменную для суммы с начальным значением 0

sum = sum + x1; //прибавляем первое слагаемое

sum = sum + x2; //прибавляем второе слагаемое

sum = sum + x3; //прибавляем третье слагаемое

return sum; //возвращаем результат

Заголовок функции: int Sum(int x1, int x2, int x3) начинается с указания типа значения, возвращаемого функцией. Поскольку функция находит сумму трех переменных типа int, то и возвращаемое значение принадлежит тому же типу.

Int – тип данных. Служит для представления целых чисел.

В качестве имени функции выбрано содержательное имя, - для суммы это обычно Sum или Summa. Имя указано в латинице, хотя можно использовать и имя, заданное кириллицей, например, - Сумма. Латиница предпочтительнее, поскольку не требует частых переключений с одного алфавита на другой при работе с программой.

1. Обработчик событий.

private void buttonSum\_Click(object sender, EventArgs e)

{

int value1, value2, value3;

value1 = int.Parse(textBoxValue1.Text);

value2 = int.Parse(textBoxValue2.Text);

value3 = int.Parse(textBoxValue3.Text);

int sum = Sum(value1, value2, value3);

textBoxMessage.Text = sum.ToString();

}

Имя процедуры, представляющей обработчик события, строится из имени кнопки – buttonSum и имени события – Click.

В обработчике события объявляются три переменные. Значения эти переменные получают в соответствии с данными, прочитанными из окошек. Обратите внимание, в текстовых окнах *пользователь* задает значения как текст. Поэтому данные, представляющие целые числа, должны быть преобразованы, - из типа string приведены к типу int. Это преобразование выполняет метод Parse, разбирающий строку, преобразуя ее в число. Конечно, в строке должно быть записано число, чтобы такое преобразование могло выполниться без ошибок.

После ввода необходимых данных вызывается приведенная выше *функция* Sum, вычисляющая сумму значений. Результат помещается в окно сообщений. Все выполняется в полном соответствии с приведенным выше описанием работы обработчика событий.

**Функции**

int Max(int x1, int x2, int x3)

{

int max = int.MinValue; //вводим переменную для максимума с начальным значением MinValue

if (x1 > max) max = x1; //максимум из одного значения

if (x2 > max) max = x2; //максимум из двух значений

if (x3 > max) max = x3; //максимум из трех значения

return max; //возвращаем результат

}

void MaxAndName(int x1, int x2, int x3,

string name1, string name2, string name3,

out int max, out string name\_max)

{

max = int.MinValue; //вводим переменную для максимума с начальным значением MinValue

name\_max = "";

if (x1 > max)

{

max = x1; name\_max = name1; //максимум из одного значения и имя

}

if (x2 > max)

{

max = x2; name\_max = name2; //максимум из двух значений и имя

}

if (x3 > max)

{

max = x3; name\_max = name3; //максимум из трех значений и имя

}

}

**Обработчики**

private void buttonMax\_Click(object sender, EventArgs e)

{

int value1, value2, value3;

value1 = int.Parse(textBoxValue1.Text);

value2 = int.Parse(textBoxValue2.Text);

value3 = int.Parse(textBoxValue3.Text);

int max = Max(value1, value2, value3);

textBoxMessage.Text = max.ToString();

}

private void buttonMaxName\_Click(object sender, EventArgs e)

{

//входные данные. Ввод значений

int value1, value2, value3;

string name1, name2, name3;

value1 = int.Parse(textBoxValue1.Text);

value2 = int.Parse(textBoxValue2.Text);

value3 = int.Parse(textBoxValue3.Text);

name1 = textBoxName1.Text;

name2 = textBoxName2.Text;

name3 = textBoxName3.Text;

//результаты

int max\_value;

string max\_name;

//вычисления

MaxAndName(value1, value2, value3,

name1, name2, name3,

out max\_value, out max\_name);

//вывод результатов

textBoxMessage.Text = max\_name + " собрал(а) грибов " +

max\_value + ". Больше всех!";

}

**Задание 2. Задача о соревнованиях по триатлону**

В соревнованиях по триатлону спортсменам нужно проплыть 1500 метров, проехать на велосипеде 40 километров и пробежать 10 километров. Из трех спортсменов Андрей лучше всех плавает, Егор быстрее бегает, а Михаил лучший на велосипеде. Известны скорости этих трех спортсменов в каждом виде состязаний. Определите время, затраченное участниками на прохождение дистанции, и кто из них стал победителем.

Для облегчения работы приведу возможный интерфейс проекта, предназначенного для решения этой задачи:

