

Министерство науки и высшего образования Российской Федерации  
Чайковский филиал  
федерального государственного автономного образовательного учреждения  
высшего образования  
**Пермский национальный исследовательский  
политехнический университет**

## **МЕТОДИЧЕСКИЕ ПРЕДПИСАНИЯ**

**к выполнению курсовой работы**

по дисциплине

*«Программирование и алгоритмизация»*

Направление: 15.03.04 Автоматизация технологических процессов и производств

Очная форма обучения  
Очно-заочная форма обучения  
Заочная форма обучения

Сухих, И.И. Методические предписания к выполнению курсовой работы по дисциплине «Программирование и алгоритмизация» для всех форм обучения направления подготовки 15.03.04 Автоматизация технологических процессов и производств.

Составитель: ст. преподаватель кафедры автоматизации, информационных и инженерных технологий ЧФ ПНИПУ Сухих И.И.

Методические предписания к выполнению курсовой работы по дисциплине «Программирование и алгоритмизация» для всех форм обучения направлений подготовки: 15.03.04 Автоматизация технологических процессов и производств. Методические предписания составлены в соответствии с рабочими программами дисциплины.

Методические предписания обсуждены и одобрены на заседании кафедры «Автоматизации, информационных и инженерных технологий» ЧФ ПНИПУ «Об» декабря 2021 г., протокол №14

Методические предписания к выполнению лабораторных работ рекомендованы методической комиссией ЧФ ПНИПУ для использования в учебном процессе (протокол № 4 от 30.12.2021)

©Пермский национальный исследовательский  
политехнический университет  
Чайковский филиал, 2021  
©Сухих И.И., 2021

## СОДЕРЖАНИЕ

Цель и задачи курсовой работы .....	4
Порядок выполнения курсовой работы.....	4
Структура курсовой работы .....	5
Задания на курсовую работу .....	7
Методические предписания по работе со средой разработки Visual Studio.....	8
Методические предписания по оформлению курсовой работы .....	16
Приложения .....	22

## **1 Цель и задачи курсовой работы**

Курсовая работа по дисциплине «Программирование и алгоритмизация» выполняется в соответствии с учебным планом направления подготовки 15.03.04 Автоматизация технологических процессов и производств. Дисциплина посвящена построению систем автоматизации и управления, проектированию алгоритмов и разработке программного обеспечения.

Цель курсовой работы – получение студентом знаний об объектно-ориентированном программировании, знаний о взаимодействии элементов пользовательского интерфейса; приобретение практических навыков разработки приложений; применение средств разработки приложений; использование С-подобного языка программирования.

## **2 Порядок выполнения**

Курсовая работа выполняется в соответствии с заданием, выдаваемым каждому студенту. Задание на курсовую работу выдается преподавателем в сроки, установленные учебным расписанием. В задании на курсовую работу указывается тема, перечень подлежащих разработке вопросов и обязательного графического материала, исходные данные к работе и рекомендованная литература. А также указывается дата выдачи задания преподавателем, дата получения задания студентом и дата защиты курсовой работы. Задание на курсовую работу подписывается студентом и преподавателем. Варианты заданий на курсовую работу представлены в приложении А.

Курсовая работа выполняется каждым студентом индивидуально под руководством преподавателя. При выдаче задания на курсовую работу преподаватель разъясняет цели и задачи работы, излагает требования к содержанию и оформлению пояснительной записки и графического материала. Выполненная и оформленная курсовая работа представляется преподавателю не позднее, чем за неделю до защиты. Преподаватель проверяет соответствие пояснительной записки и графических материалов заданию, корректность и обоснованность принятых технических решений и расчетов, грамотность и логичность изложения материала пояснительной записки, правильность выполнения графического материала. Правильно выполненная и оформленная курсовая работа допускается к защите. Если курсовая работа содержит принципиальные ошибки, недоделки или не соответствует заданию, то она возвращается студенту на доработку. Исправленная курсовая работа представляется преподавателю для повторной проверки. Защита курсовой работы производится в установленные сроки. Защита курсовой работы состоит из доклада на 10-12 минут с использованием мультимедиа проектора и ответов на вопросы

преподавателя. В докладе должна быть полностью отражена тема курсовой работы. По результатам защиты выставляется оценка. При выставлении оценки учитывается:

- научно-технический уровень курсовой работы и глубина исследования;
- качество выполнения пояснительной записки и графического материала;
- четкость и логичность изложения доклада, правильность ответов на вопросы;
- степень самостоятельности при работе над курсовой работой.

### **3 Структура курсовой работы**

Курсовая работа по дисциплине Программирование и алгоритмизация состоит из трех частей:

Теоретическая часть: теоретические сведения о предмете исследования.

Практическая часть: практическая реализация поставленной цели в техническом задании.

**Первая часть** курсовой работы заключается в поиске, анализе и обработке информации по указанной теме и представляет собой анализ о предмете исследования. Должна содержать сведения, актуальные на сегодняшний день и описывающие предмет исследования с позиций классификации, индивидуальных особенностей, принципов функционирования и применения в составе автоматизированной системы.

**Вторая часть** курсовой работы является практической частью и заключается реализации разработанных алгоритмов решения поставленной перед студентом задачи. Разработка и отладка приложения на С-подобном языке программирования с использованием современных средств.

Пояснительная записка курсовой работы по дисциплине Программирование и алгоритмизация включает:

- Титульный лист.
- Задание на курсовую работу.
- Содержание.
- Введение.
- Теоретические сведения о предмете исследования.
- Сведения, требуемые для решения поставленной задачи.
- Реализация разработанных алгоритмов на С-подобном языке программирования.
- Заключение.

- Список литературы.
- Приложения.

К пояснительной записке обязательно прилагается задание на курсовую работу, которое не включаются в содержание и общее количество страниц пояснительной записки.

### **Титульный лист**

Титульный лист должен содержать наименование темы, фамилию и инициалы студента, наименование группы, фамилию и инициалы преподавателя, год выполнения работы. Пример оформления титульного листа приведен в приложении Б.

### **Введение**

В начале данного раздела нужно обосновать выбор темы, определяемый ее актуальностью, формулируется проблема и круг вопросов, необходимых для решения. Описывается актуальность исследования. Указывается актуальность работы. В конце раздела формулируются цель курсовой работы и задачи, которые необходимо решить для достижения поставленной цели. Объем «Введения» должен быть 1 - 2 страницы.

**Теоретические сведения о предмете исследования** Раздел должен содержать краткие сведения о предмете исследования, включающую в себя историческую справку о выбранном языке программирования, современные сведения средства разработки. Необходимо построить подробный алгоритм решения поставленной задачи. Описать конструктивные особенности и характеристики предмета исследования. Выявить сильные и слабые стороны. Сделать сравнительный анализ. В конце раздела требуется подвести итог проделанному анализу. Описание предмета исследования следует сопроводить иллюстративным материалом: рисунками, схемами, графиками и таблицами. Объем раздела с учетом иллюстративного материала не должен превышать 15 страниц.

### **Сведения, требуемые для решения поставленной задачи**

Раздел должен содержать классификацию возможных вариантов реализации предмета исследования. Описываются варианты реализации, которые возможно использовать для решения поставленной задачи. В конце раздела требуется сделать вывод о надежности предмета исследования. Определить способы оптимизации. Описание неисправностей следует сопроводить иллюстративным материалом: рисунками, схемами, графиками и таблицами. Объем раздела с учетом иллюстративного материала не должен превышать 10 страниц.

### **Реализация разработанных алгоритмов на С-подобном языке программирования**

Раздел должен содержать формулировку задания. Блок-схема разработанного

алгоритма. Описание процессов, протекающих в ходе выполнения алгоритма. Описание способов программирования. Описание приложения, а также описание всех ключевых блоков программы. Иллюстрация работы программы. Листинг программ приводится в приложении.

В конце раздела требуется сделать вывод по полученным результатам. Объем раздела с учетом иллюстративного материала не должен превышать 10 страниц.

### **Заключение**

В заключение излагаются выводы и предложения, полученные студентом в процессе выполнения курсовой работы. Выводы пишутся в тезисном стиле (по пунктам). В конце заключения необходимо включить фразу «Цели и задачи курсовой работы достигнуты».

### **Список литературы**

Включает список используемых источников в количестве не менее 10. В отчете необходимо привести ссылки на литературу. Список литературы оформляется согласно первому упоминанию источника в отчете по курсовой работе.

### **Приложения**

В приложения включают вспомогательный материал и данные, не вынесенные в основную часть, но являющиеся необходимыми при реализации задачи курсовой работы. Например: схемы, графики, объемные таблицы, исходный код программы, дополнительные иллюстрации.

## **4 Задания на курсовую работу**

В теоретической и практической части курсовой работы необходимо произвести исследование современных автоматизированных рабочих мест по видам деятельности, построить разноаспектную классификацию, выявить конструктивные особенности и характеристики. На основе анализа определить сильные и слабые стороны. Список примерных тем:

- Разработка АРМ библиотекаря для автоматизации процесса учета выдачи книг в библиотеке.
- Разработка АРМ диспетчера такси для автоматизации процесса управления пассажироперевозками.
- Разработка АРМ сотрудника кадрового агентства для автоматизации процесса подбора вакансий.
- Разработка АРМ кассира кинотеатра для автоматизации процесса продаж билетов.
- Разработка АРМ кассира парикмахерской для автоматизации процесса

предоставления услуг парикмахерской.

- Разработка АРМ кассира для автоматизации процесса управления кассовым аппаратом.
- Разработка АРМ компании грузоперевозок для автоматизации процесса предоставления услуг грузоперевозки.
- Разработка АРМ сотрудника мастерской по ремонту мобильных телефонов парикмахерской для автоматизации процесса предоставления услуг.
- Разработка АРМ продажи билетов в ЖД кассе для автоматизации процесса продаж билетов.
- Разработка АРМ продажи билетов в ЖД кассе для автоматизации процесса продаж билетов.
- Разработка АРМ врача-терапевта для автоматизации сбора, обработки и хранения информации о пациенте.
- Разработка АРМ врача-стоматолога для автоматизации расчетов стоимости услуг.
- Разработка АРМ управления продуктовой базы.
- Моделирование работы интернет-магазина.
- Учет заявок на ремонт техники.
- Учет документов на предприятии.

## **5 Методические предписания по работе со средой разработки Visual Studio**

Среда Visual Studio визуально реализуется в виде одного окна с несколькими панелями инструментов. Количество, расположение, размер и вид панелей может меняться программистом или самой средой разработки в зависимости от текущего режима работы среды или пожеланий программиста, что значительно повышает производительность работы.

При запуске Visual Studio появляется начальная страница со списком последних проектов, а также командами *Создать проект* и *Открыть проект*. Нажмите ссылку *Создать проект* или выберите в меню *Файл* команду *Создать проект*, на экране появится диалог для создания нового проекта (рис. 1.1).

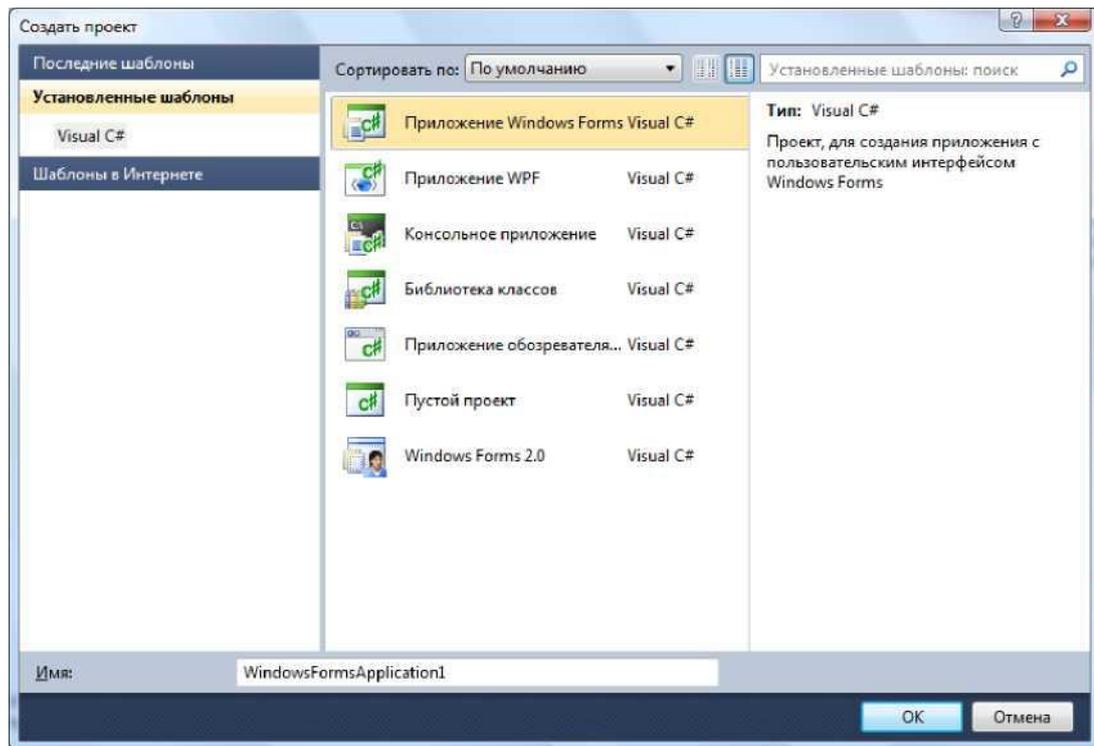


Рис. 1.1. Диалог создания нового проекта

Слева в списке шаблонов приведены языки программирования, которые поддерживает данная версия Visual Studio: убедитесь, что там выделен раздел Visual C#. В средней части приведены типы проектов, которые можно создать. В наших лабораторных работах будут использоваться два типа проектов:

1. *Приложение Windows Forms* - данный тип проекта позволяет создать полноценное приложение с окнами и элементами управления (кнопками, полями ввода и пр.) Такой вид приложения наиболее привычен большинству пользователей.

2. *Консольное приложение* - в этом типе проекта окно представляет собой текстовую консоль, в которую приложение может выводить тексты или ожидать ввода информации пользователя. Консольные приложения часто используются для вычислительных задач, для которых не требуется сложный или красивый пользовательский интерфейс.

Выберите в списке тип проекта «Приложение Windows Forms», в поле «имя» внизу окна введите желаемое имя проекта и нажмите кнопку ОК. Через несколько секунд Visual Studio создаст проект и Вы сможете увидеть на экране картинку, подобную представленной на рис. 1.2.

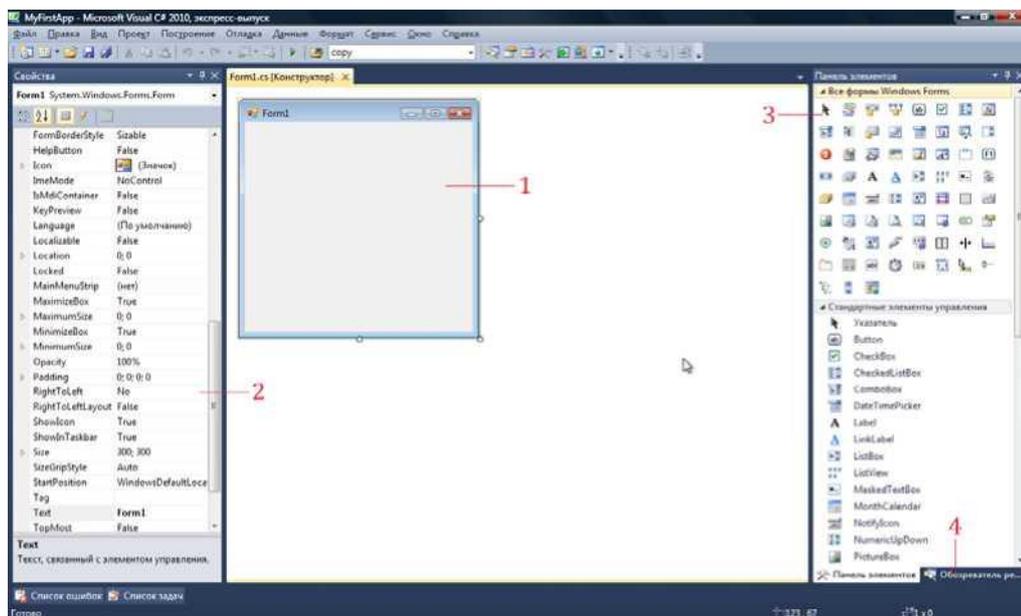


Рис. 1.2. Главное окно Visual Studio

В главном окне Visual Studio присутствует несколько основных элементов, которые будут помогать нам в работе. Прежде всего, это **форма (1)** - будущее окно нашего приложения, на котором будут размещаться элементы управления. При выполнении программы помещенные элементы управления будут иметь тот же вид, что и на этапе проектирования.

Второй по важности объект - это **окно свойств (2)**, в котором приведены все основные свойства выделенного элемента управления или окна. С помощью кнопки можно просматривать свойства элемента управления, а кнопка переключает окно в режим просмотра событий. Чтобы было удобнее искать нужные свойства, можно отсортировать их по алфавиту с помощью кнопки . Если этого окна на экране нет, его можно активировать в меню *Вид ^ Окно свойств* (иногда этот пункт вложен в подпункт *Другие окна*).

Сами элементы управления можно брать на **панели элементов (3)**. Все элементы управления разбиты на логические группы, что облегчает поиск нужных элементов. Если панели нет на экране, ее нужно активировать командой *Вид ^ Панель элементов*.

Наконец, **обозреватель решений (4)** содержит список всех файлов, входящих в проект, включая добавленные изображения и служебные файлы. Активируется командой *Вид ^ Обзорщик решений*.

Указанные панели могут уже присутствовать на экране, но быть скрытыми за другими панелями или свернуты к боковой стороне окна. В этом случае достаточно

щелкнуть на соответствующем ярлычке, чтобы вывести панель на передний план.

**Окно текста** программы предназначено для просмотра, написания и редактирования текста программы. Переключаться между формой и текстом программы можно с помощью команд *Вид ^ Код* и *Вид ^ Конструктор*. При первоначальной загрузке в окне текста программы находится текст, содержащий минимальный набор операторов для нормального функционирования пустой формы в качестве Windows-окна. При помещении элемента управления в окно формы, текст программы автоматически дополняется описанием необходимых для его работы библиотек стандартных программ (раздел *using*) и переменных для доступа к элементу управления (в скрытой части класса формы).

Программа на языке C# составляется как описание алгоритмов, которые необходимо выполнить, если возникает определенное событие, связанное с формой (например, щелчок «мыши» на кнопке - событие *Click*, загрузка формы - *Load*). Для каждого обрабатываемого в форме события, с помощью окна свойств, в тексте программы организуется метод, в котором программист записывает на языке C# требуемый алгоритм.

### **Настройка формы**

Настройка формы начинается с настройки размера формы. С помощью мыши, «захватывая» одну из кромок формы или выделенную строку заголовка, отрегулируйте нужные размеры формы.

Для настройки будущего окна приложения задаются свойства формы. Для задания любых свойств формы и элементов управления на форме используется окно свойств.

Новая форма имеет одинаковые имя (*Name*) и заголовок (*Text*) - *Form1*.

Для изменения заголовка щелкните кнопкой мыши на форме, в окне свойств найдите и щелкните мышкой на строчке с названием *Text*.

### **Размещение элементов управления на форме**

Для размещения различных элементов управления на форме используется панель элементов. Панель элементов содержит элементы управления, сгруппированные по типу. Каждую группу элементов управления можно свернуть, если она в настоящий момент не нужна. Для выполнения лабораторных работ потребуются элементы управления из группы *Стандартные элементы управления*.

Щелкните на элементе управления, который хотите добавить, а затем щелкните в нужном месте формы - элемент появится на форме. Элемент можно перемещать по форме, схватившись за него левой кнопкой мышки (иногда это можно сделать лишь за появляющийся при нажатии на элемент квадрат со стрелками)). Если элемент управления

позволяет изменять размеры, то на соответствующих его сторонах появятся белые кружки, ухватившись за которые и можно изменить размер. После размещения элемента управления на форме, его можно выделить щелчком мыши и при этом получить доступ к его свойствам в окне свойств.

### Размещение строки ввода

Если необходимо ввести из формы в программу или вывести на форму информацию, которая вмещается в одну строку, используют окно однострочного редактора текста, представляемого элементом управления `TextBox`.

В данной программе с помощью однострочного редактора будет вводиться имя пользователя.

Выберите на панели элементов пиктограмму с названием `TextBox`, щелкните мышью в том месте формы, где вы хотите ее поставить. Захватив его мышкой, отрегулируйте размеры элемента управления и его положение. Обратите внимание на то, что теперь в тексте программы можно использовать переменную `textBox1`, которая соответствует добавленному элементу управления. В этой переменной в свойстве `Text` будет содержаться строка символов (тип `string`) и отображаться в соответствующем окне `TextBox`.

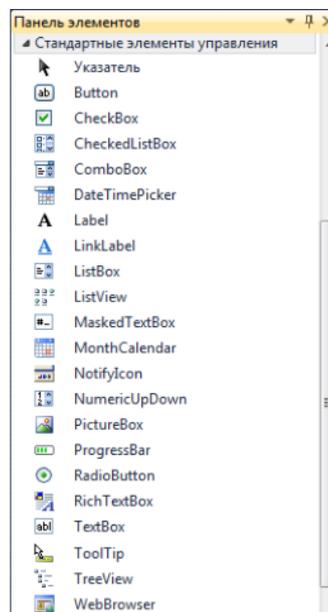


Рис. 1.3. Панель элементов

С помощью окна свойств установите шрифт и размер символов, отражаемых в строке `TextBox` (свойство `Font`).

### Размещение надписей

На форме могут размещаться пояснительные надписи. Для нанесения таких надписей

на форму используется элемент управления Label. Выберите на панели элементов пиктограмму с названием Label, щелкните на ней мышью. После этого в нужном месте формы щелкните мышью, появится надпись label1. Щелкнув на ней мышью, отрегулируйте размер и, изменив свойство Text в окне свойств, введите строку, например «Введите свое имя:», а также выберите размер символов (свойство Font).

Обратите внимание, что в тексте программы теперь можно обращаться к новой переменной типа Label. В ней хранится пояснительная строка, которую можно изменять в процессе работы программы.

### **Написание программы обработки события**

С каждым элементом управления на форме и с самой формой могут происходить события во время работы программы. Например, с кнопкой может произойти событие - нажатие кнопки, а с окном, которое проектируется с помощью формы, может произойти ряд событий: создание окна, изменение размера окна, щелчок мыши на окне и т. п. Эти события могут обрабатываться в программе. Для обработки таких событий необходимо создать обработчики события - специальный *метод*. Для создания обработчика события существует два способа.

Первый способ - создать обработчик для события по умолчанию (обычно это самое часто используемое событие данного элемента управления). Например, для кнопки таким образом создается обработчик события нажатия.

### **Написание программы обработки события нажатия кнопки**

Поместите на форму кнопку, которая описывается элементом управления Button. С помощью окна свойств измените заголовок (Text) на слово «Привет» или другое по вашему желанию. Отрегулируйте положение и размер кнопки.

После этого два раза щелкните мышью на кнопке, появится текст программы:

```
private void button1_Click(object sender, EventArgs e)
{
}
```

Это и есть обработчики события нажатия кнопки. Вы можете добавлять свой код между скобками { }. Например, наберите:

```
MessageBox.Show("npueem, " + textBox1.Text + "!");
```

### **Написание программы обработки события загрузки формы**

Второй способ создания обработчика события заключается в выборе соответствующего события для выделенного элемента на форме. При этом используется

окно свойств и его закладка . Рассмотрим этот способ. Выделите форму щелчком по ней, чтобы вокруг нее появилась рамка из точек. В окне свойств найдите событие Load. Щелкните по данной строчке дважды мышкой. Появится метод:

```
private void Form1_Load(object sender, EventArgs e)
{
}
```

Каждый элемент управления имеет свой набор обработчиков событий, однако некоторые из них присущи большинству элементов управления. Наиболее часто применяемые события описаны ниже:

- **Activated**: форма получает это событие при активации.
- **Load**: возникает при загрузке формы. В обработчике данного события следует задавать действия, которые должны происходить в момент создания формы, например установка начальных значений.
- **KeyPress**: возникает при нажатии кнопки на клавиатуре. Параметр `e.KeyChar` имеет тип `char` и содержит код нажатой клавиши (клавиша Enter клавиатуры имеет код #13, клавиша Esc - #27 и т. д.). Обычно это событие используется в том случае, когда необходима реакция на нажатие одной из клавиш.
- **KeyDown**: возникает при нажатии клавиши на клавиатуре. Обработчик этого события получает информацию о нажатой клавише и состоянии клавиш `Shift`, `Alt` и `Ctrl`, а также о нажатой кнопке мыши. Информация о клавише передается параметром `e.KeyCode`, который представляет собой перечисление `Keys` с кодами всех клавиш, а информацию о клавишах-модификаторах `Shift` и др. можно узнать из параметра `e.Modifiers`.
- **KeyUp**: является парным событием для `KeyDown` и возникает при отпускании ранее нажатой клавиши.
- **Click**: возникает при нажатии кнопки мыши в области элемента управления.
- **DoubleClick**: возникает при двойном нажатии кнопки мыши в области элемента управления.

### **Запуск и работа с программой**

Запустить программу можно, выбрав в меню *Отладка* команду *Начать отладку*. При этом происходит трансляция и, если нет ошибок, компоновка программы и создание единого загружаемого файла с расширением `.exe`. На экране появляется активное окно

программы.

Если в программе есть ошибки, то в окне Список ошибок появятся найденные ошибки, а программа обычно не запускается. Иногда, однако, может быть запущена предыдущая версия программы, в которой еще нет последних изменений: чтобы этого не происходило, нужно в настройках Visual Studio установить опцию показа окна ошибок и запретить запуск при наличии ошибок (рис. 1.4 и 1.5).

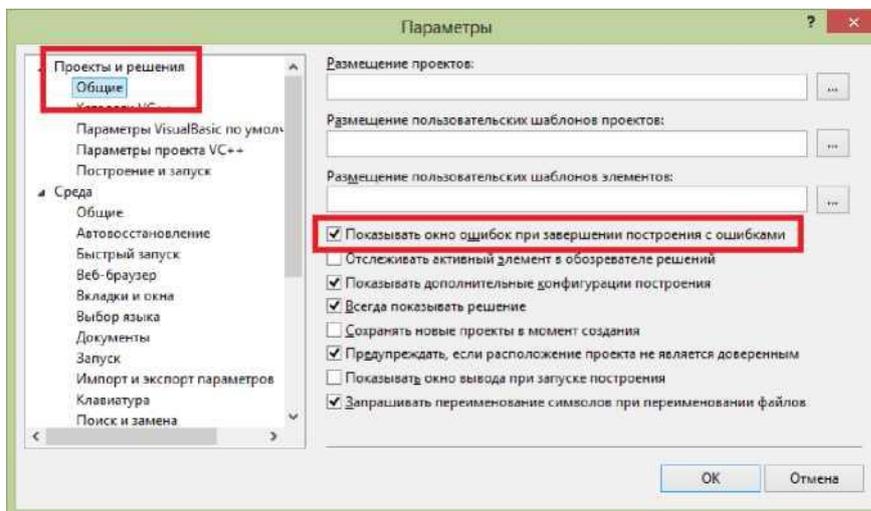


Рис. 1.4. Опция показа сообщений об ошибках

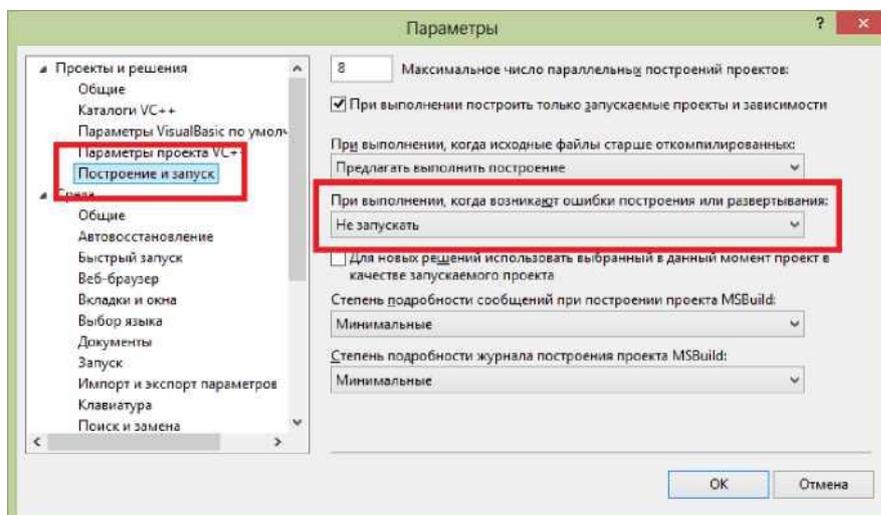


Рис. 1.5. Опция отключения запуска предыдущей версии при ошибках построения

Для завершения работы программы и возвращения в режим проектирования формы не забудьте закрыть окно программы!

#### Динамическое изменение свойств

Свойства элементов на окне могут быть изменены динамически во время выполнения программы. Например, можно изменить текст надписи или цвет формы. Изменение свойств происходит внутри обработчика события (например, обработчика

события нажатия на кнопку). Для этого используют оператор присвоения вида:

```
<имя элемента>.<свойство> = <значение>;
```

Например:

```
label1.Text = "Привет";
```

<Имя элемента> определяется на этапе проектирования формы, при размещении элемента управления на форме. Например, при размещении на форме ряда элементов TextBox, эти элементы получают имена textBox1, textBox2, textBox3 и т. д., которые могут быть заменены в окне свойств в свойстве (Name) для текущего элемента. Допускается использование латинских или русских символов, знака подчеркивания и цифр (цифра не должна стоять в начале идентификатора). Список свойств для конкретного элемента можно посмотреть в окне свойств, а также в приложении к данным методическим указаниям.

## **5 Методические предписания по оформлению курсовой работы**

Пояснительная записка должна быть напечатана на одной стороне листа бумаги формата А4, кегль – 14, шрифт – "Times New Roman", интервал полуторный, форматирование по ширине страницы. Переносы в пояснительной записке не допускаются. Размеры полей: левое – 30 мм, правое – 15 мм, верхнее и нижнее поля – 20 мм каждое. Расстояние между заголовком и последующим текстом не менее 6 пт; между заголовком раздела и подраздела – 3 пт. Интервал между текстом предыдущего раздела или подраздела и заголовком последующего должен быть не меньше 6 пт. Абзацы в тексте начинаются отступом, равным 1.25 см. Разделы и подразделы должны иметь порядковые номера, обозначенные арабскими цифрами без точки и записанные с абзацного отступа. Заголовки глав, пунктов и подпунктов должны быть краткими и должны точно отражать содержание раздела. Заголовки глав, пунктов и подпунктов записывают в виде предложения с абзацного отступа, с прописной буквы, без точки в конце. Переносы слов в заголовках не допускаются. Точка в конце заголовков не ставится. Главы записки рекомендуется начинать с нового листа. Нумерация страниц начинается с титульного листа и заканчивается последним листом пояснительной записки. Номер страницы проставляется в центре нижней части листа без точки. На титульном листе номер страницы не указывается.

### **Оформление содержания**

Содержание включает все пронумерованные заголовки. Наименования, включенные в содержание, записываются, начиная с прописной буквы. Для каждого

названия указывается номер страницы, на которой оно находится. Не следует использовать в содержании сокращения, кроме общепринятых обозначений.

### Нумерация разделов записки

Главы нумеруются в пределах пояснительной записки порядковыми номерами арабскими цифрами. Пункты нумеруются по порядку в пределах главы, а подпункты нумеруются в пределах пункта. Введение, заключение и список использованных источников не нумеруются. *Пример:* 4.2 – (пункт 2 главы 4). 2.1.1 – (подпункт 1 пункта 1 главы 2).

### Оформление рисунков

Иллюстрации могут быть выполнены как черно-белом, так и в цветном исполнении. Рисунки нумеруются в пределах главы. *Пример:* Рисунок 3.1 – (рисунок первый в третьей главе). Рисунок может содержать: название, поясняющие надписи, если они необходимы, расположенные под рисунком и номер рисунка под поясняющей надписью.

Номер рисунка и его название располагаются под рисунком или под пояснительными надписями посередине строки. Слово «Рисунок» записывается полностью, с прописной буквы без точки после номера. Название рисунка записывается в продолжении строки через дефис с прописной буквы. Пример оформления приведен на рисунке 3.1.

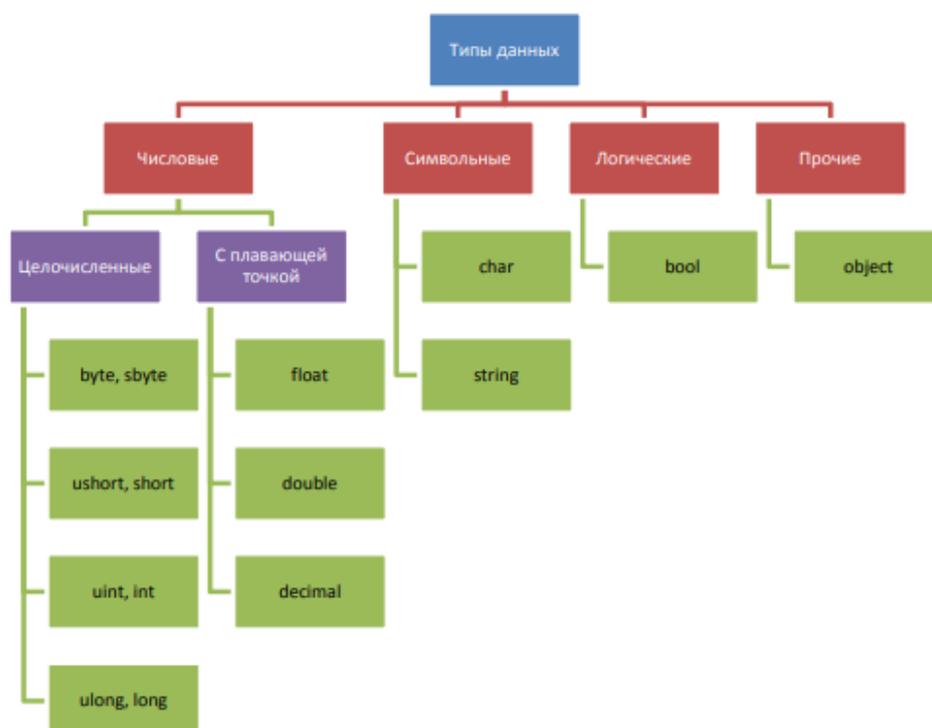


Рисунок 3.1 – Структура типов данных

Если рисунок располагается на нескольких листах, то на каждом последующем листе указывается номер рисунка, за которым следует слово "Продолжение". *Пример:* Рисунок 3.1 Продолжение

При ссылках на иллюстрации следует писать «...в соответствии с рисунком 1» (при сквозной нумерации) или «... в соответствии с рисунком 4.1» (при нумерации в пределах раздела). Сокращения слова «рисунок» не допускается. Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. *Пример:* Рисунок А.3 – Блок-схема

### **Оформление таблиц**

Таблицы нумеруются арабскими цифрами. Например, «Таблица 1» или «Таблица 3.1» (первая таблица в разделе (главе) 3). Таблицы приложения обозначаются аналогично, но с добавлением перед цифрой обозначения приложения. *Пример:* Таблица Б.1 – (таблица 1 приложения Б).

Таблица может иметь заголовок, который размещается над таблицей слева без абзацного отступа в одну строку с ее номером через тире и записывается строчными буквами (кроме первой прописной). Таблицы выравниваются по ширине окна. Общий вид таблицы рассмотрим на примере таблицы 3.1.

Таблица 3.1 – Общий вид таблицы


Таблицу помещается под текстом, в котором впервые дана ссылка на неё. Допускается применять размер шрифта в таблице меньший, чем в тексте, но не менее 8.

Таблица может располагаться на нескольких листах. Заголовок таблицы указывается только на первом листе и нижняя граница таблицы не проводится. На последующих листах кроме последнего в правом верхнем углу пишется "Продолжение таблицы ..." и указывается номер. Точка после номера не ставится. *Пример:* Продолжение таблицы 3.1

Продолжение таблицы 3.1


На последнем листе, где заканчивается таблица в правом верхнем углу пишется "Окончание таблицы ..." и указывается номер. Например: Окончание таблицы 3.1


### **Оформление приложений**

Иллюстрации, таблицы или тексты вспомогательного характера допускается оформлять в виде приложений. На все приложения в тексте работы должны быть даны ссылки. Приложения располагаются в порядке ссылок на них в тексте. Каждое приложение начинается с новой страницы, с указанием наверху посередине страницы слова «Приложение», его обозначения. Приложение должно иметь заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой. Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ. После слова —Приложение следует буква, обозначающая его последовательность. Приложение должно иметь заголовок, который пишется симметрично относительно текста с прописной буквы отдельной строкой. Если в отчете одно приложение, то оно обозначается так: Приложение А. Текст каждого приложения, при необходимости, может быть разделен на разделы, подразделы, пункты, подпункты, которые нумеруют в пределах каждого приложения. Перед номером ставится обозначение этого приложения. Рисунки каждого приложения и таблицы обозначаются отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Например, Рисунок А.5 или Таблица Б.2.

### **Оформление ссылок**

В пояснительной записке должны быть ссылки на все рисунки, таблицы, формулы, приложения, литературные источники, которые приведены в записке. Рисунки, таблицы, формулы располагаются сразу после появления первой ссылки на них, то есть на текущем или следующем листе записки. Порядок номеров приложений и литературных ссылок должен соответствовать порядку появления ссылок на них. При ссылке на литературный источник указывается его порядковый номер, заключенный в квадратные скобки. Например, [4] или [4,5,6]. Ссылка на источник должна ставиться в той части предложения, где начинается пересказываемая мысль другого автора. В обзорах целесообразно использовать ряд устойчивых речевых штампов, например: «...как утверждается в [...]...», «более подробно с...можно познакомиться в [ ]», «В основу этой методики положены идеи, высказанные в [...]» и т.п. Ссылка ставится при первом упоминании материала из источника.

При первой ссылке на рисунок пишется, например, «... в соответствии с рисунком

1.4» или (рисунок 1.4). При первой ссылке на таблицу пишется, например, в таблице 2.3 или (таблица 2.3). Ссылки в тексте на номер формулы дают в скобках, например, ... в формуле (2.5). При ссылке на приложение пишется полностью слово "приложение" и указывается его обозначение, например, "...в приложении А" или (приложение А).

#### **Описание книги четырех и более авторов**

Описание книги начинается с заглавия, если она написана четырьмя и более авторами. Авторы пишутся только в сведениях об ответственности. При необходимости их количество сокращают. Также дается описание коллективных монографий, сборников статей. *Примеры:*

Язык программирования С# / А. Хейлмберг, М. Торгерсен, С. Вилтамут, П. Голд – СПб.: Питер, 2012. – 784 с.

Автомобильный транспорт: научная монография; [под ред. Р.М. Ахмеднабиева]. Новосибирск: Изд. «СибАК», 2013. – 168 с.

#### **Описание статьи из журнала**

Для статьи из журнала нужно указать авторов статьи, название статьи, название журнала, год, номер выпуска и страницы начала и окончания статьи. *Пример.*

Морозов Е.А. Об устойчивости интегральных кривых в сопряженных пространствах. // Вестник ИжГТУ. – 2005. – №3 – С. 39 – 41.

Ефимов И.Н., Морозов Е.А., Селиванов К.М., Ермолаева Е.В. Каноническое преобразование фазового пространства в динамике твердого тела // Вестник ИжГТУ. – 2009. – №4 – С. 190 – 195.

#### **Учебники, учебные пособия**

Для учебников и учебных пособий нужно указать: авторов, название, отметка о типе издания учебное пособие или учебник, город и название издательства, год издания, количество страниц. *Примеры:*

Нетушил А.В. Теория автоматического управления: Нелинейные системы, управление при случайных воздействиях: учебник для вузов. – 2-е изд., перераб. и доп. – М.:Высш. школа, 1983. – 432с.

Богословский С.Г., А.Д. Дорофеев. Динамика полета летательных аппаратов: Учебное пособие. – СПб.: СПбГУАП, 2002. – 64 с.

#### **Описание электронных изданий**

Для электронных источников нужно указать: авторов, название статьи, название сайта (или раздела сайта) и адрес URL. Также должна присутствовать дата обращения к источнику и отметка [Электронный ресурс]. *Примеры:*

Закирова Э.И. Управление образовательными системами с использованием мультимедийных технологий [Электронный ресурс]. Электронное научно-техническое издание «Наука и образование». URL: <http://technomag.bmstu.ru/doc/606440.html> (дата обращения: 10.09.2021).

Твердотельный накопитель // Википедия. URL: <http://ru.wikipedia.org/?oldid=69875253> (дата обращения: 10.04.2021).

## ПРИЛОЖЕНИЕ А

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Чайковский филиал  
федерального государственного автономного  
образовательного учреждения высшего профессионального образования  
«Пермский национальный исследовательский политехнический университет»  
(ЧФ ПНИПУ)

### ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ по дисциплине Программирование и алгоритмизация

Ф.И.О. студента \_\_\_\_\_  
Специальность, индекс группы \_\_\_\_\_

Наименование темы: « \_\_\_\_\_ »

#### 1. Область применения

АРМ предназначено для \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

#### 2. Цели и задачи

Основной целью курсовой работы является получение студентом знаний об объектно-ориентированном программировании, знаний о взаимодействии элементов пользовательского интерфейса; приобретение практических навыков разработки приложений; применение средств разработки приложений; использование С-подобного языка программирования.

Основные задачи:

- 2.1. Проектирование структуры приложения
- 2.2. Проектирование интерфейса пользователя
- 2.3. Реализация ввода и редактирования данных
- 2.4. Реализация вывода информации

#### 3. Состав разрабатываемого приложения

3.1 Приложение должно обеспечивать:

— \_\_\_\_\_;  
— \_\_\_\_\_;  
— \_\_\_\_\_;

3.2 Состав информации, которая хранится в приложении:

— \_\_\_\_\_;  
— \_\_\_\_\_;  
— \_\_\_\_\_;

#### 4. Используемые технические и программные средства

Минимальные требования:

4.1 Персональный компьютер:

Процессор:

Тип: совместимый с Pentium Dual Core или выше;

Быстродействие: не менее 2,0 ГГц рекомендуется 2.5 ГГц и выше;

Память:

Оперативная память: не менее 2 ГБ, рекомендуется 3 ГБ и более;

- Жесткий диск: не менее 100 ГБ;
- 4.2 Монитор: не менее 15 дюймов и с разрешением 1024 x 768 пикселей;
- 4.3 Операционная система: Windows 7 SP1 и выше;
- 4.4 Среда разработки: Microsoft Visual Studio 2015 Community и выше.

### **5. Подсистемы, реализуемые в ходе курсовой работы**

- 5.1 Проектирование логической структуры АРМ;
- 5.2 Сбор/эмуляция информации для заполнения справочников (справочники должны содержать не менее 5 - 10 записей);
- 5.3 Разработка стандартных форм функций;
- 5.4 Разработка клиентского приложения.

### **6. Состав документации**

- 6.1 Пояснительная записка;
- 6.2 Текст программы (модулей) с комментариями;
- 6.3 Контрольный пример;
- 6.4 Руководство пользователя.

Срок представления работы (проекта) к защите « \_\_\_\_\_ » \_\_\_\_\_ 202 г.

Руководитель работы (проекта) \_\_\_\_\_  
(подпись, дата) (фамилия, имя, отчество)

Задание принял к исполнению \_\_\_\_\_  
(подпись, дата) (фамилия, имя, отчество)

## ПРИЛОЖЕНИЕ Б

Министерство науки и высшего образования Российской Федерации  
Чайковский филиал  
Федерального государственного автономного образовательного учреждения  
высшего образования

Пермский национальный исследовательский  
политехнический университет

Кафедра автоматизации, информационных и инженерных технологий

### **Курсовая работа** *«название работы»*

по дисциплине  
*«Программирование и алгоритмизация»*

Выполнил студент группы \_\_\_\_\_

\_\_\_\_\_

Проверил ст. преподаватель:

Сухих И.И.