

Приложение 4
к рабочей программе дисциплины
МДК.01.01 Разработка программных модулей

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»

Димитровградский инженерно-технологический институт –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(ДИТИ НИЯУ МИФИ)

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

**по организации внеаудиторной самостоятельной работы студентов
по дисциплине**

МДК.01.01 Разработка программных модулей

специальности 09.02.07 Информационные системы и программирование

Составитель: А.В. Надеждина, преподаватель техникума ДИТИ НИЯУ
МИФИ

Димитровград

СОДЕРЖАНИЕ

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2 ПРОГРАММА ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ	5
3 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ	6

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время актуальным становятся требования к личным качествам современного студента – умению самостоятельно пополнять и обновлять знания, вести самостоятельный поиск необходимого материала, быть творческой личностью. Ориентация учебного процесса на саморазвивающуюся личность делает невозможным процесс обучения без учета индивидуально-личностных особенностей обучаемых, предоставления им права выбора путей и способов обучения. Появляется новая цель образовательного процесса – воспитание личности, ориентированной на будущее, способной решать типичные проблемы и задачи исходя из приобретенного учебного опыта и адекватной оценки конкретной ситуации.

Решение этих задач требует повышения роли самостоятельной работы студентов над учебным материалом, усиления ответственности преподавателя за развитие навыков самостоятельной работы, за стимулирование профессионального роста студентов, воспитание их творческой активности и инициативы.

Введение в практику учебных программ и модулей с повышенной долей самостоятельной работы активно способствует модернизации учебного процесса.

Методические рекомендации для внеаудиторной самостоятельной работы (ВСР) по МДК.01.01 Разработка программных модулей, предназначены для студентов второго и третьего курса специальности 09.02.07 Информационные системы и программирование.

Внеаудиторная самостоятельная работа студентов является обязательной для каждого студента, определяется учебным планом.

Основными целями внеаудиторной самостоятельной работы студентов являются:

- овладение знаниями и умениями по специальности;
- формирование готовности к самообразованию, самостоятельности и ответственности;
- развитие творческого подхода к решению проблем учебного и профессионального уровня.

Выполнение студентами ВСР способствует формированию общих компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

Выполнение студентами ВСР способствует формированию профессиональных компетенций:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

Выполнение студентами ВСР способствует формированию воспитательных компетенций:

В17 Формирование чувства личной ответственности за научно-технологическое развитие России, за результаты исследований и их последствия

В18 Формирование ответственности за профессиональный выбор, профессиональное развитие и профессиональные решения

В19 Формирование научного мировоззрения, культуры поиска нестандартных научно-технических решений, критического отношения к исследованиям лженаучного толка

В22 Формирование профессиональной ответственности в области разработки модулей программного обеспечения для компьютерных систем, осуществления интеграции программных модулей, разработки, администрирования и защиты баз данных, сопровождения и обслуживания программного обеспечения компьютерных систем

В23 Формирование творческого мышления и стремления к постоянному самосовершенствованию

В24 Формирование профессионально значимых установок: не производить, не копировать и не использовать программные и технические средства, не приобретённые на законных основаниях; не нарушать признанные нормы авторского права; не нарушать тайны передачи сообщений, не практиковать вскрытие информационных систем и сетей передачи данных; соблюдать конфиденциальность доверенной информации

**2 ПРОГРАММА ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ
МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ**

Наименование разделов и тем дисциплины	Объем, часов	Коды формируемых компетенций	Виды СРС	Формы контроля СРС	Сроки выполнения
1	2	3	4	5	6
Выполнение разработки спецификаций отдельных компонентов					
Раздел 1. Формирование алгоритмов разработки программных модулей в соответствии с техническим заданием					
Тема 1.3. Структурное программирование	2	ОК 01, ОК 02, ОК 09, ПК 1.1	Решение задач с использованием Pascal	индивидуальная проверка	к текущему занятию
Раздел 2. Разработка программных модулей с помощью Delphi в соответствии с техническим заданием					
Тема 2.2. Событийно-управляемое программирование	1	ОК 01, ОК 02, ОК 09, ПК 1.1, ПК 1.2, В17-19, В22-В24	Рассмотреть и сделать отчет по Базовым компонентам панели Standard	индивидуальная проверка	к текущему занятию
	1	ОК 01, ОК 02, ОК 09, ПК 1.1, ПК 1.2, В17-19, В22-В24	Создать форму с логическими элементами управления	индивидуальная проверка	к текущему занятию
	2	ОК 01, ОК 02, ОК 09, ПК 1.1, ПК 1.2, В17-19, В22-В24	Рассмотреть и сделать отчет по Базовым компонентам панелей Additional, Win32, System и Dialogs.	индивидуальная проверка	к текущему занятию
Тема 2.3. Графические возможности Delphi	2	ОК 01, ОК 02, ОК 09, ПК 1.1, ПК 1.2, В17-19, В22-В24	Решение задач с использованием Delphi	индивидуальная проверка	к текущему занятию
Раздел 3. Разработка программных модулей с помощью Visual C# в соответствии с техническим заданием					
Тема 3.2. Объектная ориентация языка C#	1	ОК 01, ОК 02, ОК 09, ПК 1.1, ПК 1.2, В17-19, В22-В24	Решение задач с использованием C#.	индивидуальная проверка	к текущему занятию
Тема 3.10. Методы C#	2	ОК 01, ОК 02, ОК 09, ПК 1.1, ПК 1.2, В17-19, В22-В24	Решение задач с использованием C#.	индивидуальная проверка	к текущему занятию

3 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельные работы выполняются индивидуально на домашнем компьютере или в компьютерном классе в свободное от занятий время.

Студент обязан:

- перед выполнением самостоятельной работы, повторить теоретический материал, пройденный на аудиторных занятиях;
- выполнить работу согласно заданию;
- по каждой самостоятельной работе представить преподавателю отчет в виде результирующего файла на внешнем носителе;
- ответить на поставленные вопросы.

При выполнении самостоятельных работ студент должен сам принять решение об оптимальном использовании возможностей программного обеспечения. Если по ходу выполнения самостоятельной работы у студентов возникают вопросы и затруднения, он может консультироваться у преподавателя. Каждая работа оценивается по пятибалльной системе. Критерии оценки приведены в конце методических рекомендаций.

Тема 1.3. Структурное программирование

Кол-во часов - 2

Практический блок:

Написать программу использующую модуль, в котором описаны 2 процедуры: первая находит периметр треугольника по введенным сторонам, а вторая площадь по введенным сторонам.

Цель работы: систематизировать знания, полученные при создании программы.

Примерный план:

Пользуясь языком программирования Pascal, написать программу из двух модулей. Будьте готовы отвечать на вопросы по исследуемой теме.

Тема 2.2. Событийно-управляемое программирование

Кол-во часов - 1

Практический блок:

Рассмотреть и сделать отчет по Базовым компонентам панели Standard: Label, Edit, Button, CheckBox, RadioButton, ComboBox, ListBox, Memo. Использовать пособие Н. Культина «Основы программирования в Delphi XE». Создать кнопки перехода от формы к форме.

Цель работы: систематизировать знания, полученные при повторении компонентов.

Примерный план:

Пользуясь средой разработки Delphi, создать проект и формы с данными компонентами (рис.1). Готовиться к тестированию.

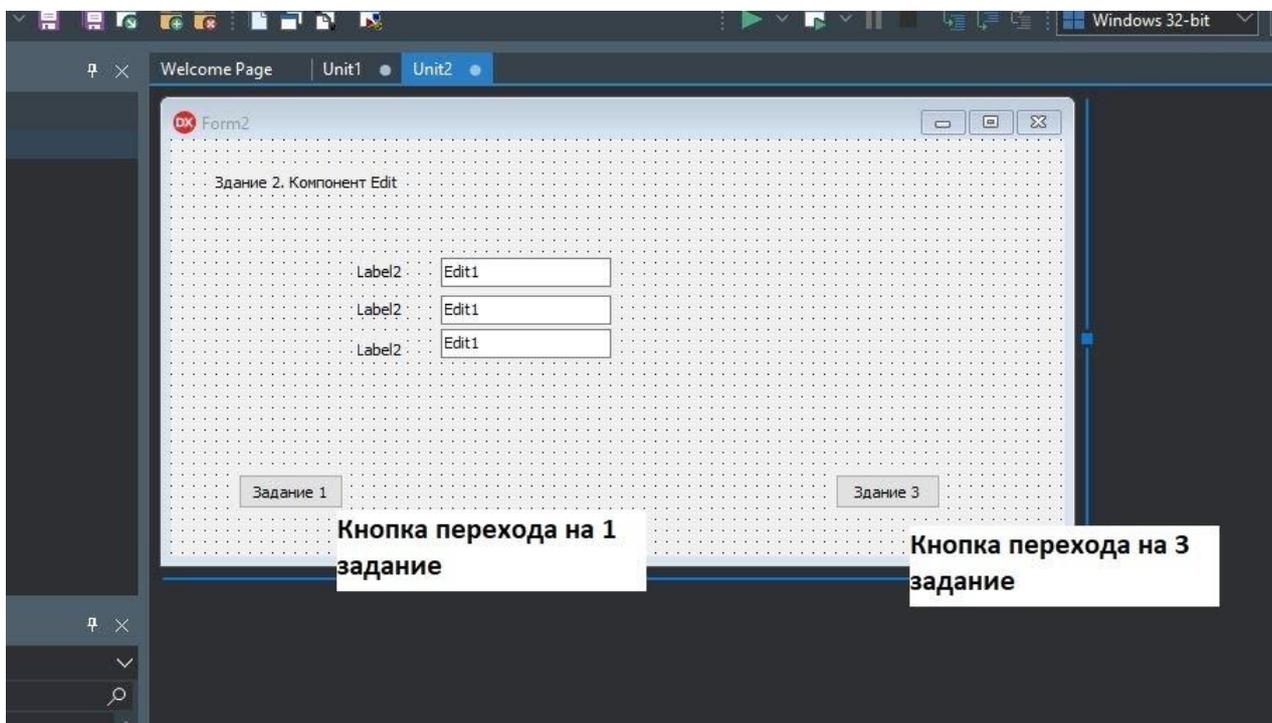


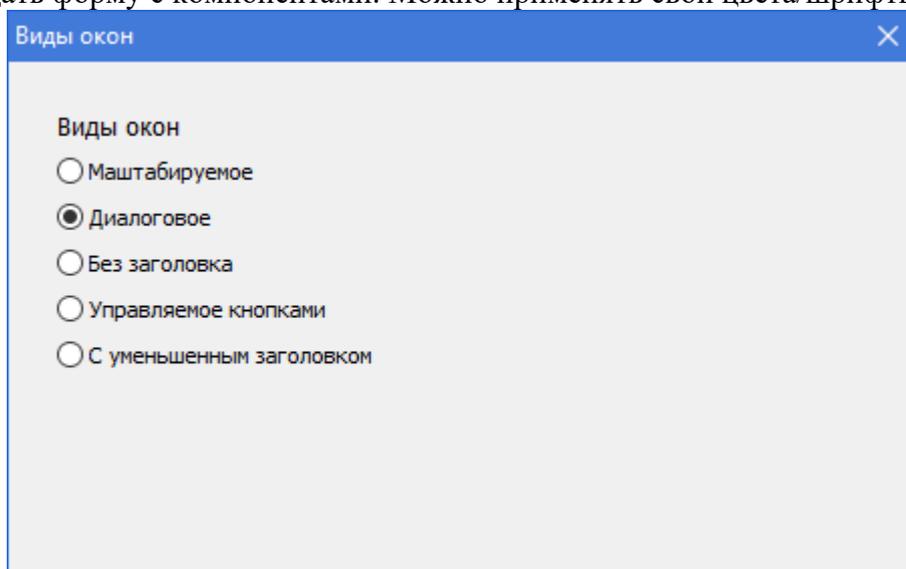
Рисунок 1 – Пример задания с компонентом Edit

Тема 2.2. Событийно-управляемое программирование

Кол-во часов - 1

Практический блок:

1. Создать форму с компонентами. Можно применять свои цвета/шрифты.



Цель работы: систематизировать знания, полученные при создании программы.

Примерный план:

Пользуясь средой разработки Delphi, создать проект и форму с компонентами. Будьте готовы отвечать на вопросы по исследуемой теме.

Тема 2.2. Событийно-управляемое программирование

Кол-во часов - 2

Практический блок:

Рассмотреть и сделать отчет по Базовым компонентам панели Additional, Win32, System и Dialogs: Timer, Panel, ControlBar, SpeedButton, StatusBar, UpDown, ProgressBar, Image, MainMenu, OpenFileDialog, SaveDialog. Использовать пособие Н. Культина «Основы программирования в Delphi XE». Создать кнопки перехода от формы к форме.

Цель работы: систематизировать знания, полученные при повторении компонентов.

Примерный план:

Пользуясь средой разработки Delphi, создать проект и формы с данными компонентами (рис.1). Готовиться к тестированию.

Тема 2.3. Графические возможности Delphi

Кол-во часов - 2

Практический блок:

1. Написать программу, показывающую два разных движущихся смайла Emodji.
2. Написать программу, показывающую тонущий кораблик. На экране должна отображаться линия океана. Кораблик должен плыть слева направо, на середине формы, кораблик должен двигаться вниз (т.е. тонуть).

Цель работы: систематизировать знания, полученные при создании программы.

Примерный план:

Пользуясь средой разработки Delphi, создать проект, форму с компонентами, написать программный код. Будьте готовы отвечать на вопросы по исследуемой теме.

Тема 3.2. Объектная ориентация языка C#

Кол-во часов - 1

Практический блок:

Задание 1:

Выполнить: Создайте C# консольное приложение. В функции Main() приложения создайте переменные с верным типом данных (используйте информацию, представленную ниже). После объявления и инициализации переменных, выведите значения этих переменных в консоль, используйте метод `Console.WriteLine()`.

Цель работы: систематизировать знания, полученные при повторении компонентов.

Примерный план:

Пользуясь средой разработки Visual Studio, создать консольное приложение.

Информация о студенте:

Имя | Фамилия | Дата рождения | Адрес | Пол | Страна

Информация о курсе:

Указание: Для указания пола студента необходимо использовать символьный тип данных:
`char gender='m';`

Результат:

```
Информация о студенте:  
Имя : Ivan, Фамилия: Ivanov, Дата рождения: 1993 1 2,  
Адрес: Rostov-on-Don, Пол: m, Страна: Russia  
Course Information:  
Название курса: CS211, Баллы:56, Количество занятий в неделю: 16,  
Преподаватель: Mayer Svetlana
```

Тема 3.10. Методы C#

Кол-во часов - 1

Практический блок:

Выполнить: Создайте метод `Minmax()` который принимает два целочисленных аргумента по ссылке (`ref`) и меняет их значения таким образом, что первый становится максимальным, а второй — минимальным (меняет значения аргументов, если требуется). Создайте перегруженный метод `Minmax` для трех параметров.

Цель работы: систематизировать знания, полученные при повторении компонентов.

Примерный план:

Указание 1: Необходимо использовать ключевое слово `static` в сигнатуре создаваемого метода, т.к. функция `Main` тоже `static`, невозможно вызвать не `static` метод из `static` метода.

Указание 2: Метод не возвращает значения значит, используется ключевое слово `void` в сигнатуре).

Пример выполнения:

```
Введите два числа  
2 4  
Результат после метода Minmax : 4 2  
Введите три числа  
2 4 3  
Результат после перегрузки метода Minmax : 4 3 2
```

Алгоритм:

1. Создайте консольное приложение, назовите проект `Metod1`.
2. В окне `Solution Explorer` (Обозреватель решений) найдите файл `Program.cs` и переименуйте его в `SamMetod1.cs`.
3. В теле функции `Main` попросите пользователя ввести два числа. Присвойте введенные значения переменным:

```
...  
Console.WriteLine("Введите два числа");  
int a = int.Parse(Console.ReadLine());  
int b = int.Parse(Console.ReadLine());  
...
```

4. Затем, то же самое сделайте для перегруженной функции с тремя аргументами. Попросите пользователя ввести три числа и присвойте введенные значения переменным:

```
...
Console.WriteLine("Введите три числа");
a = int.Parse(Console.ReadLine());
b = int.Parse(Console.ReadLine());
int c = int.Parse(Console.ReadLine());
...
```

5. Для добавления метода установите курсор после тела функции `Main()`.
6. Объявите метод `Minmax()`, который будет менять значения переменных, если это потребуется. Используйте модификатор `ref` для аргументов:

```
...
static void Minmax(ref int a, ref int b)
{
    ...
}
...
```

Модификатор `ref` используется для того, чтобы значение аргумента передалось из основной программы в создаваемую функцию и, наоборот, из функции измененное значение должно вернуться в основную программу. *Ref* аргументы должны быть инициализированы до их вызова, иначе возникнет ошибка. При вызове модификатор `ref` также указывается.

7. Найдите максимальное и минимальное значения, используя стандартные функции `Min()` и `Max()`. Присвойте найденные значения соответствующим переменным (первый аргумент должен быть больше второго):

```
static void Minmax(ref int a, ref int b)
{
    int max = Math.Max(a, b);
    int min = Math.Min(a, b);
    a = max;
    b = min;
    ...
}
```

8. Поскольку созданный метод `Minmax` не возвращает никакого значения, то новые значения переменных будут передаваться в основную программу, благодаря модификатору `ref`.
9. Теперь можно вызвать созданный метод из функции `Main`:

```
...
Minmax(ref a, ref b);
Console.WriteLine($"Результат после метода Minmax: {a}, {b}");
...
```

10. Запустите приложение и проверьте результат.
11. Чтобы свернуть или развернуть код какого-либо метода, можно использовать горячие клавиши `Ctrl+M` (дважды).

Перегруженный метод (функция):

12. Установите курсор после метода `Minmax()` и создайте перегруженный метод (с таким же именем) для трех аргументов:

```
...
static void Minmax(ref int a, ref int b, ref int c)
{
    Minmax(ref a, ref b);
    Minmax(ref b, ref c);
    Minmax(ref a, ref b);
}
...
```

В коде метода найдем сначала минимальное и максимальное из переменных `a` и `b`, затем из `b` и `c`, и после — из `a` и `b` еще раз, так как значения переменных могли уже поменяться к этому моменту.

13. Наконец добавьте вызов метода из функции Main():

```
...  
Minmax(ref a,ref b, ref c);  
Console.WriteLine($"After overloaded Minmax the result is: {a}, {b}, {c}");  
...
```

14. Запустите приложение и проверьте результат. Несмотря на то, что имена у созданных методов одинаковы, компилятор вызовет нужный метод, исходя из количества аргументов.