

**Методические указания по проведению практических занятий
и организации СРС
по дисциплине «Программная инженерия»
(направления 09.03.03 «Прикладная информатика»)**

Введение

Предлагаемые методические указания разработаны в соответствии с рабочей программой дисциплины «Программная инженерия» направления подготовки 09.03.03 «Прикладная информатика». Методические указания предназначены для использования студентами, изучающими данную дисциплину, на практических занятиях и в ходе самостоятельной работы.

В ходе выполнения практических заданий и самостоятельной работы студенты должны получить следующие навыки и умения:

- работа в команде;
- анализ требований заказчика к создаваемым программным системам;
- формулировка функциональных требований к создаваемым программным системам;
- разрабатывать диаграммы вариантов использования, последовательности и деятельности на языке UML;
- оптимизировать программный код.

Все практические задания рассчитаны на работу группы студентов как единой системы. Каждое задание делится на небольшие части, выполнение которых осуществляется малыми подгруппами (2-4 человека). В ходе выполнения задачи регулярно проводятся круглые столы, на которых обсуждаются основные части проекта, разработанные разными подгруппами. Главным требованием к студентам является согласованность результатов выполненной работы.

Практическое задание 1

Проанализировать приведенные ниже требования заказчика к системе «Информационно-образовательная среда». Данная среда предназначена для обмена необходимой информацией между студентами и преподавателями образовательного учреждения. Она включает в себя возможность добавления и просмотра лекционных и методических материалов, а так же наличие простейших средств коммуникации. На основе требований разработать функциональные требования к системе. Возможно разделение требований на тематические блоки, например: общие требования к системе, требования для администрирования системы, требования для группы пользователей «Студент», и т.д. Обратите внимание на то, что требования заказчика изложены не формализовано, в связи с чем допускаются различные толкования некоторых пунктов. Все спорные моменты необходимо обсудить с заказчиком (преподавателем).

Требования заказчика:

1. Наличие таких групп пользователей как: студенты, преподаватели и администраторы.
2. Наличие личного кабинета.
3. Преподаватель должен иметь доступ ко всем дисциплинам.
4. Преподаватель должен уметь работать с материалами дисциплин, которые он ведет.
5. Преподаватель должен менять информацию о дисциплинах.
6. Студент может просматривать дисциплинам своей специальности.
7. Студент может общаться с преподавателями.
8. Администратор должен иметь доступ к работе с дисциплинами.
9. Администратор должен иметь доступ к работе с пользователями.
- 10.Администратор определяет, кто из преподавателей ответственен за дисциплину.
- 11.Преподаватель должен иметь возможность групповой рассылки информации.

- 12.Администратор должен иметь возможность групповой рассылки информации.
- 13.Преподаватель может оставлять заявку на дисциплину.
- 14.Заявки просматриваются администраторами.
- 15.Оформление соответствует корпоративным цветам и стандартам.

Примеры функциональных требований:

1. **Аутентификация и авторизация.** Система должна предоставить возможность пользователю, вызвать модуль «Безопасность», путем нажатия на ссылку «Вход» после заполнения полей логин и пароль. При этом происходит проверка введенного логина и пароля пользователя, и в случае корректного ввода, у пользователя появляется возможность использовать права роли, к которой он относится (студент, модератор).

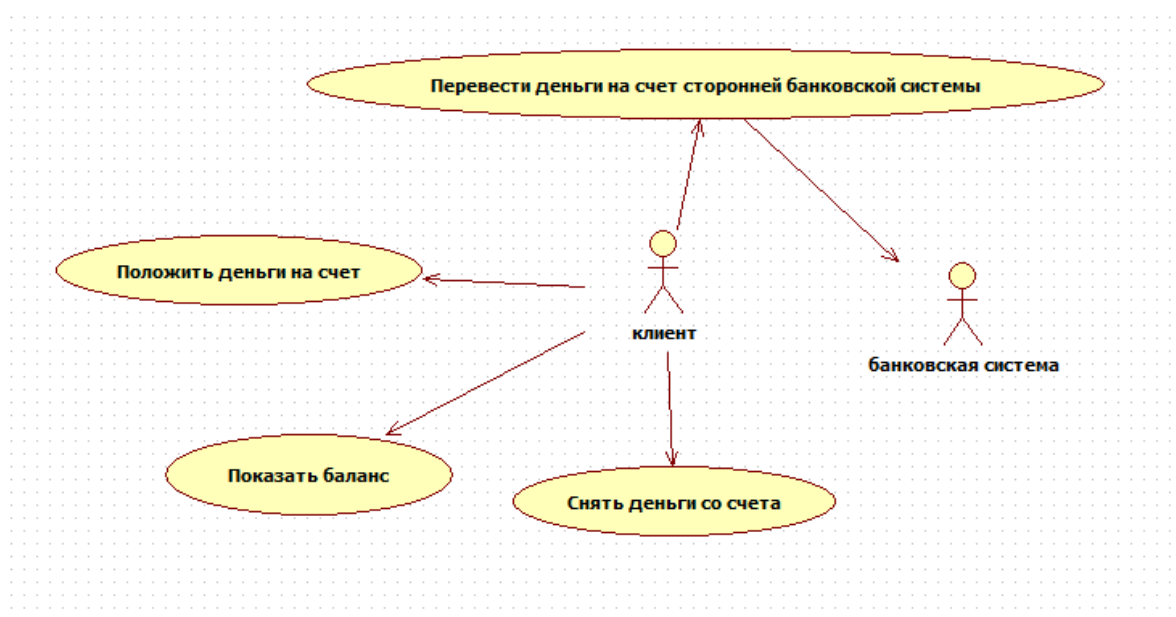
2. **Вход в личный кабинет.** Система должна предоставить возможность студенту, прошедшему аутентификацию в системе, вызвать модуль «Личный кабинет», путем нажатия на ссылку «Личный кабинет ». При этом слушатель попадает на страницу личного кабинета.

3. **Переход на главную страницу.** Система должна предоставить возможность студенту вызвать функцию перехода на главную страницу, независимо от того на какой из страниц он сейчас находится, путем перехода по ссылке «На главную» на панели личного кабинета. При этом осуществляется переход на главную страницу учебного портала, с сохранением статуса пройденной аутентификации.

Практическое задание 2

На основе составленных функциональных требований разработать диаграмму вариантов использования. Диаграммы разрабатываются индивидуально каждым студентом, после чего в формате круглого стола происходит обсуждение каждой из них и составление единой диаграммы на всю учебную группу.

Пример диаграммы вариантов использования:



Практическое задание 3

Распределить между собой все варианты использования, указанные в диаграмме вариантов использования (задание 2). Распределение вариантов использования необходимо согласовать с преподавателем. Оформить каждый вариант использования по шаблону, указанному в примере. Для каждого варианта использования необходимо указать его разработчика и ответственное лицо, с которым он был согласован.

Пример варианта использования «Подача заявки на подключение интернета»:

Описание

Данный вариант использования описывает процесс подачи заявки на подключение интернета. После заполнения клиентом формы система создает

заявку (на основе введенных данных). При этом автоматически создается личный кабинет клиента, через который клиент получает возможность отслеживать обработку заявки и изменять ее. Сценарий использования личного кабинета описан в варианте использования #02 «Работа с личным кабинетом»

Разработчик: Иванов И.И.

Согласовано с: Петров С.Л.

Действующие лица:

- Клиент;
- Интернет портал;

Предусловия

1. Клиент старше 18 лет

Основной поток

1. Клиент переходит по ссылке «Подать заявку»
2. Клиент вводит адрес (улица, № дома, № квартиры)
3. Система проверяет техническую возможность подключения
4. Клиент получает ответ системы «Техническая возможность есть»
5. Клиент выбирает тариф
6. Клиент вводит персональные данные (ФИО)
7. Клиент вводит контактную информацию (телефон, e-mail)
8. Клиент подтверждает правильность ввода и отправляет заявку
9. Подтверждает согласие на обработку персональных данных
10. Система проверяет заполнение обязательных полей
11. Система сохраняет данные и присваивает заявке id, сообщая его клиенту
12. Клиент получает уведомление о «рассмотрении» заявки
13. Заявка отправляется менеджерам по работе с клиентами
14. На e-mail клиента отправляется уведомление о создании личного кабинета и пароль к доступу в кабинет (логином является e-mail)

Альтернативные потоки.

- 3а. Клиент получает уведомление о том, что в данный момент соответствующий дом не обслуживается оператором
 - 3б. Клиенту предполагается отправить заявку на подключение дома к услуге
 - 3в. Клиент вводит персональные данные (ФИО, телефон)
 - 3г. Клиент отправляет заявку
 - 3д. Система оповещает клиента о количестве заявок на подключение данного дома
-
- 10а. В случае не заполнения обязательных полей, система выдает сообщение клиенту и возвращает его на страницу заполнения формы. При этом введенные данные остаются в полях
 - 10б. Клиент заполняет оставшиеся поля и отправляет заявку

Постусловия

1. Заявка сохранена
2. Информация о появлении новой заявки отправлена представителям отдела по работе с клиентами.

Практическое задание 4

Все разработанные в рамках практического задания 3 варианты использования распределяются между студентами группы. При распределении необходимо выполнить следующее условие: студент не может получить вариант использования разработанный им, или согласованный с ним. Распределение вариантов использования необходимо согласовать с преподавателем. На основе вариантов использования необходимо спроектировать диаграммы последовательностей и оформить по следующему образцу.

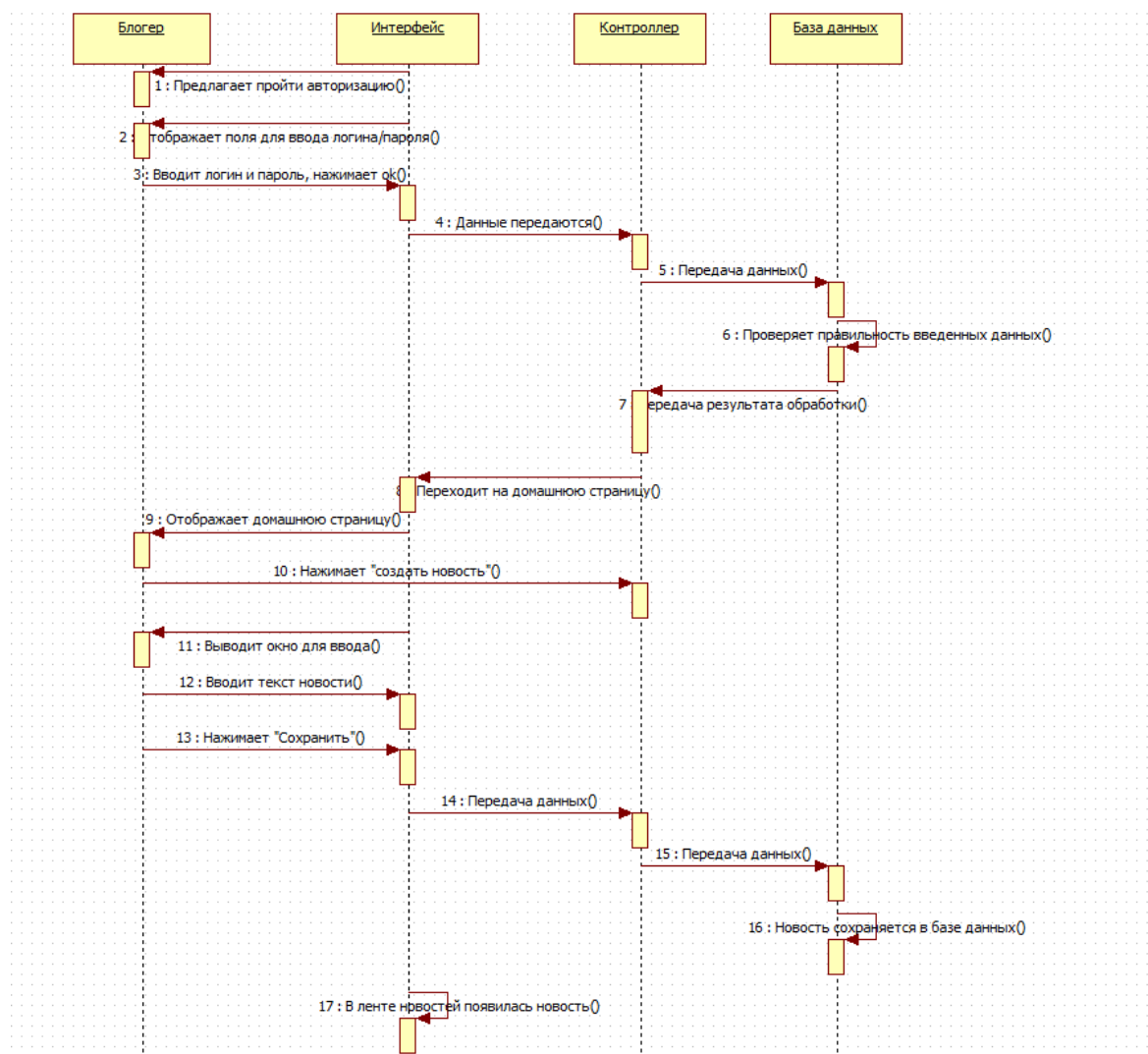
Пример диаграммы последовательностей:

Описание

Диаграмма последовательности «Снять деньги» построена на основе варианта использования «Снять деньги».

Разработчик: Иванов И.И.

Согласовано с: Петров С.Л.



Практическое задание 5

Все разработанные в рамках практического задания 3 варианты использования распределяются между студентами группы. При распределении необходимо выполнить следующее условие: студент не может получить вариант использования разработанный им, или согласованный с

ним, или вариант использования по которым им была разработана диаграмма последовательности. Распределение вариантов использования необходимо согласовать с преподавателем. На основе вариантов использования необходимо спроектировать диаграммы деятельности и оформить по следующему образцу.

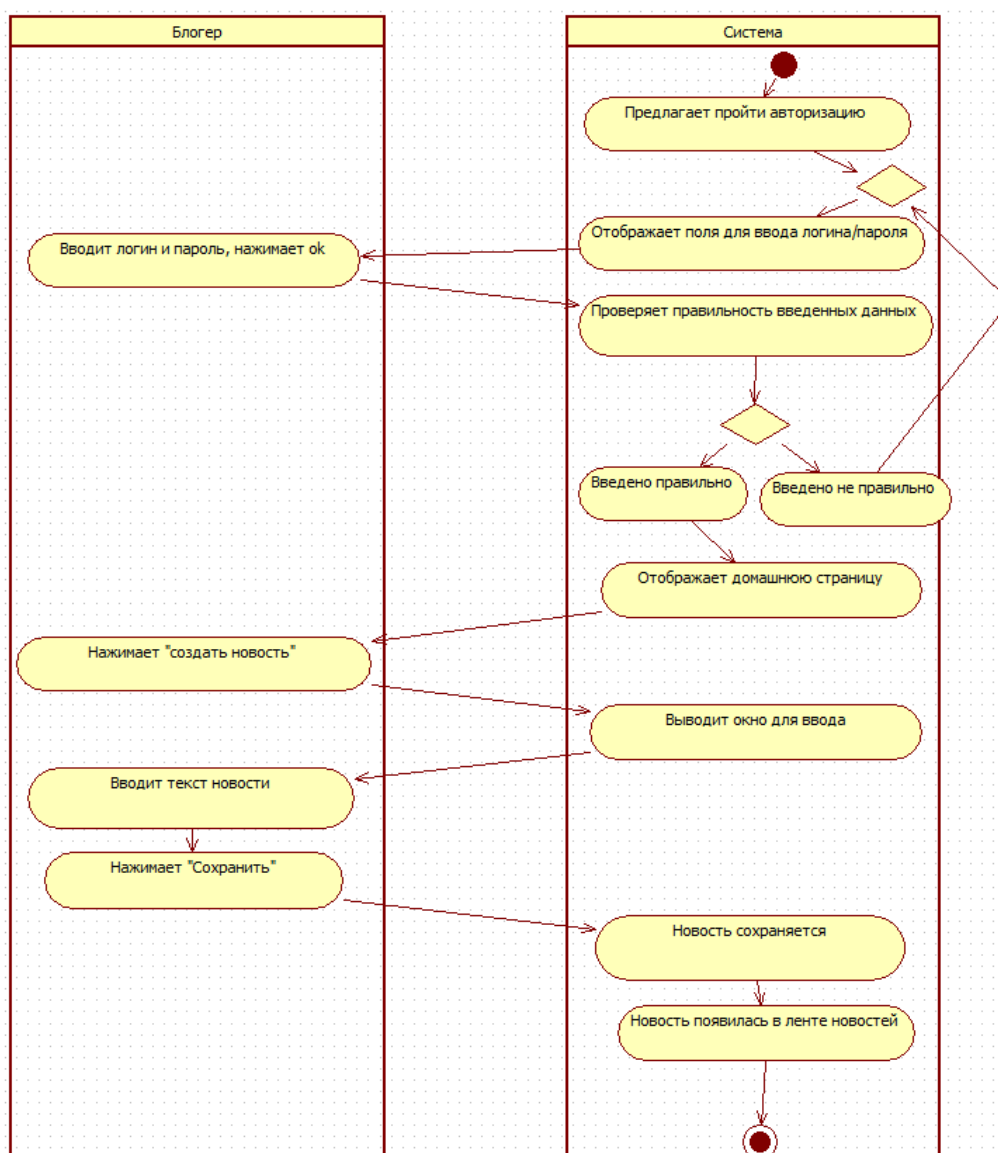
Пример диаграммы деятельности:

Описание

Диаграмма последовательности «Работа с заказом» построена на основе варианта использования «Работа с заказом».

Разработчик: Иванов И.И.

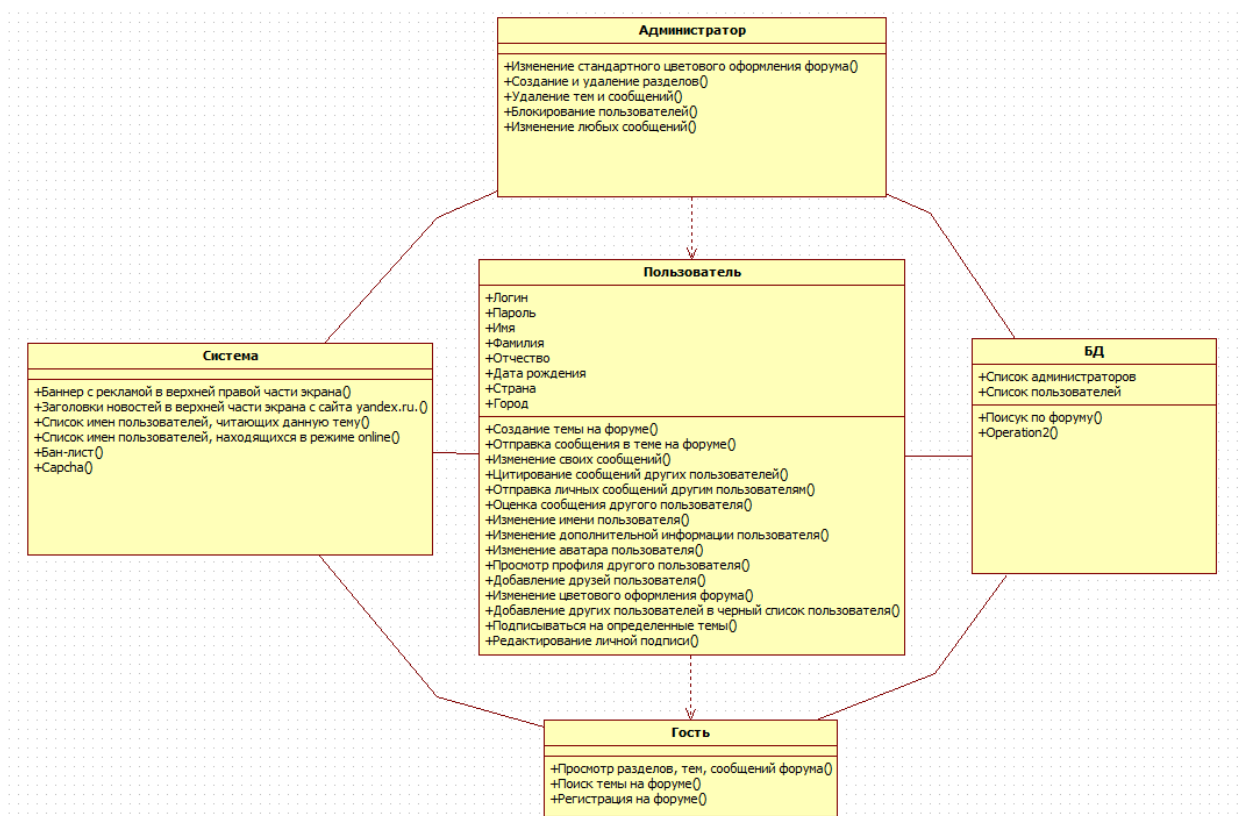
Согласовано с: Петров С.Л.



Практическое задание 6

На основе составленных функциональных требований и диаграмм (практического задания 1-5) разработать диаграмму классов. Диаграммы разрабатываются индивидуально каждым студентом, после чего в формате круглого стола происходит обсуждение каждой из них и составление единой диаграммы на всю учебную группу.

Пример диаграммы классов:



Практическое задание 7

Студентам необходимо выбрать по одному из предлагаемых ниже вариантов программного кода и согласовать свой выбор с преподавателем. Для каждого варианта необходимо выполнить следующие задания:

1. Распределить программный код по строкам с использованием общепринятых отступов.
2. Выбрать одну из известных нотаций переменных и изменить имена переменных в коде в соответствии с ней.
3. Определить задачу, выполняемую кодом, и прокомментировать код.
4. Разработать оптимальный алгоритм для решения задачи.

Вариант 1

```

using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace ConsoleApplication1 {class Program{static void
Main(string[] args){const int HDFGlsdjflsd = 100;int[,] NetsCaPEnaViGaTOR =
new int[HDFGlsdjflsd, HDFGlsdjflsd];Random Private = new Random();for (int
Hedgehock = 0; Hedgehock < HDFGlsdjflsd; Hedgehock++){ for (int YarnNEEdLE
= 0; YarnNEEdLE < HDFGlsdjflsd;
YarnNEEdLE++){NetsCaPEnaViGaTOR[Hedgehock, YarnNEEdLE] =
Private.Next() % 1000;Console.Write("{0} ", NetsCaPEnaViGaTOR[Hedgehock,
YarnNEEdLE]);}Console.WriteLine();}for (int SkySCAKJGF = 0; SkySCAKJGF
< HDFGlsdjflsd; SkySCAKJGF++){int ManuFACTurer = 0;for (int SelfContIOUS
= 0; SelfContIOUS < HDFGlsdjflsd; SelfContIOUS++){if (SelfContIOUS == 0 ||
SelfContIOUS == HDFGlsdjflsd - 1) ManuFACTurer +=
NetsCaPEnaViGaTOR[SkySCAKJGF,
SelfContIOUS];Console.WriteLine(ManuFACTurer);}}}}

```

Вариант 2

```

using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace ConsoleApplication1 {class Program{static void
Main(string[] args){const int NetsCaPEnaViGaTOR = 123;int[]
SelfContIOUSManuFACTurer = new int[NetsCaPEnaViGaTOR];Random
fUryOnYourHaedMasterSeCoNdeDition = new Random(); for (int
StATuEoFlIbErTy = 0; StATuEoFlIbErTy < NetsCaPEnaViGaTOR;
StATuEoFlIbErTy++){SelfContIOUSManuFACTurer[StATuEoFlIbErTy] =
fUryOnYourHaedMasterSeCoNdeDition.Next() % 768;Console.Write("{0} ",
SelfContIOUSManuFACTurer[StATuEoFlIbErTy]);}int fUryOnYourHaedMaster
= 0, MISunderStanDingprOBleM = SelfContIOUSManuFACTurer[0],
MISunDerStanDingprOBleM = 0;foreach (int NetsCaPEnaviGaTOR in
SelfContIOUSManuFACTurer){if (NetsCaPEnaviGaTOR >
MISunderStanDingprOBleM) { MISunderStanDingprOBleM =
NetsCaPEnaviGaTOR; MISunDerStanDingprOBleM = fUryOnYourHaedMaster; }

```

```
fUryOnYourHaedMaster++;} Console.WriteLine("{0} {1}",
MISunderStanDingprOBleM, MISunDerStanDingprOBleM);}}}
```

Вариант 3

```
using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace gardeNTracingSystem{class Program{static void
Main(string[] args){int MISunderStanDingprOBleM, StATuEoFlIbErTy,
FoUnDATionSyStemManaGeMent = 0;MISunderStanDingprOBleM =
Int32.Parse(Console.ReadLine());MISunderStanDingprOBleM += 7;for (int
fUryOnYourHaedMaster = 1; fUryOnYourHaedMaster <=
MISunderStanDingprOBleM; fUryOnYourHaedMaster++){StATuEoFlIbErTy =
1;for (int fUryOnYourHaedMasterSeCoNdeDitioN = 1;
fUryOnYourHaedMasterSeCoNdeDitioN <= fUryOnYourHaedMaster;
fUryOnYourHaedMasterSeCoNdeDitioN++) StATuEoFlIbErTy *=
fUryOnYourHaedMasterSeCoNdeDitioN;FoUnDATionSyStemManaGeMent +=
StATuEoFlIbErTy;}Console.WriteLine("\n{0}",
FoUnDATionSyStemManaGeMent);}}}
```

Вариант 4

```
using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace gardeNTracingSystem{class Program{static void
Main(string[] args){int MISunderStanDingprOBleM,
FoUnDATionSyStemManaGeMent = 0;MISunderStanDingprOBleM =
Int32.Parse(Console.ReadLine());MISunderStanDingprOBleM += 77;for (int
fUryOnYourHaedMaster = 1; fUryOnYourHaedMaster <=
MISunderStanDingprOBleM;
fUryOnYourHaedMaster++){FoUnDATionSyStemManaGeMent +=
fUryOnYourHaedMaster + 2*fUryOnYourHaedMaster;}
Console.WriteLine("\n{0}", FoUnDATionSyStemManaGeMent);}}}
```

Вариант 5

```
using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace gardeNTracingSystem{class Program{static void
```

```

Main(string[] args){int MISunderStanDingprOBleM, StATuEoFlIbErTy,
FoUnDATionSyStemManaGeMent = 0, SoMeThiNGvErYsTuPid =
3;MISunderStanDingprOBleM =
Int32.Parse(Console.ReadLine());MISunderStanDingprOBleM += 17;for (int
fUryOnYourHaedMaster = 1; fUryOnYourHaedMaster <=
MISunderStanDingprOBleM; fUryOnYourHaedMaster++){StATuEoFlIbErTy =
1;for (int fUryOnYourHaedMasterSeCoNdeDitioN = 1;
fUryOnYourHaedMasterSeCoNdeDitioN <= fUryOnYourHaedMaster;
fUryOnYourHaedMasterSeCoNdeDitioN++){StATuEoFlIbErTy *=
SoMeThiNGvErYsTuPid; FoUnDATionSyStemManaGeMent +=
StATuEoFlIbErTy;} Console.WriteLine("\n{0}",
FoUnDATionSyStemManaGeMent);}}

```

Вариант 6

```

using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace ConsoleApplication1 {class Program{static void
Main(string[] args){const int fUryOnYourHaedMaster = 125;int[,]
StATuEoFlIbErTy = new int[fUryOnYourHaedMaster,
fUryOnYourHaedMaster];Random FoUnDATionSyStemManaGeMent = new
Random();for (int MISunderStanDingprOBleM = 0; MISunderStanDingprOBleM <
fUryOnYourHaedMaster; MISunderStanDingprOBleM++){for (int
fUryOnYourHaedMasterSeCoNdeDitioN = 0;
fUryOnYourHaedMasterSeCoNdeDitioN < fUryOnYourHaedMaster;
fUryOnYourHaedMasterSeCoNdeDitioN++){StATuEoFlIbErTy[MISunderStanDi
ngprOBleM, fUryOnYourHaedMasterSeCoNdeDitioN] =
FoUnDATionSyStemManaGeMent.Next() % 476;Console.Write("{0} ",
StATuEoFlIbErTy[MISunderStanDingprOBleM,
fUryOnYourHaedMasterSeCoNdeDitioN]);}Console.WriteLine();}int
MISunderStanDingprOBleM = 0;for (int fUryOnYourHaedMasterSeCoNdeDitioN
= 0; fUryOnYourHaedMasterSeCoNdeDitioN < fUryOnYourHaedMaster;
fUryOnYourHaedMasterSeCoNdeDitioN++){for (int MISunderStanDingprOBleM

```

```

= 0; MISunderStanDingprOBleM < fUryOnYourHaedMaster;
MISunderStanDingprOBleM++)if (MISunderStanDingprOBleM ==
fUryOnYourHaedMasterSeCoNdeDitioN) MISunderStanDingprOBleM +=
StATuEoFlIbErTy[fUryOnYourHaedMasterSeCoNdeDitioN,
MISunderStanDingprOBleM];Console.WriteLine(MISunderStanDingprOBleM);}
}

```

Вариант 7

```

using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace ConsoleApplication1 {class Program{static void
Main(string[] args){const int fUryOnYourHaedMaster = 125;int[,]
StATuEoFlIbErTy = new int[fUryOnYourHaedMaster,
fUryOnYourHaedMaster];Random FoUnDAtionSyStemManaGeMent = new
Random();for (int MISunderStanDingprOBleM = 0; MISunderStanDingprOBleM <
fUryOnYourHaedMaster; MISunderStanDingprOBleM++){for (int
fUryOnYourHaedMasterSeCoNdeDitioN = 0;
fUryOnYourHaedMasterSeCoNdeDitioN < fUryOnYourHaedMaster;
fUryOnYourHaedMasterSeCoNdeDitioN++){StATuEoFlIbErTy[MISunderStanDi
ngprOBleM, fUryOnYourHaedMasterSeCoNdeDitioN] =
FoUnDAtionSyStemManaGeMent.Next() % 476;Console.Write("{0} ",
StATuEoFlIbErTy[MISunderStanDingprOBleM,
fUryOnYourHaedMasterSeCoNdeDitioN]);}Console.WriteLine();}int
MISunderStanDingprOBleM = 0;for (int fUryOnYourHaedMasterSeCoNdeDitioN
= 0; fUryOnYourHaedMasterSeCoNdeDitioN < fUryOnYourHaedMaster;
fUryOnYourHaedMasterSeCoNdeDitioN++)for (int MISunderStanDingprOBleM
= 0; MISunderStanDingprOBleM < fUryOnYourHaedMaster;
MISunderStanDingprOBleM++) if (MISunderStanDingprOBleM ==
fUryOnYourHaedMaster - fUryOnYourHaedMasterSeCoNdeDitioN - 1)
MISunderStanDingprOBleM +=
StATuEoFlIbErTy[fUryOnYourHaedMasterSeCoNdeDitioN,

```

```
MISunderStanDingprOBleM];Console.WriteLine(MISunderStanDingprOBleM);}}
}
```

Вариант 8

```
using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace ConsoleApplication1 {class Program {static void
Main(){string MISunderStanDingprOBleM = Console.ReadLine();char[]
StATuEoFlIbErTy = " .;!&".ToCharArray();int JDHskhadlasdh = 0;string[]
fUryOnYourHaedMaster =
MISunderStanDingprOBleM.Split(StATuEoFlIbErTy);foreach (string
fUryOnYourHaedMasterSeCoNdeDitioN in fUryOnYourHaedMaster){char
NetsCaPEnaViGaTOR = fUryOnYourHaedMasterSeCoNdeDitioN[0];for (int
SelfContIOUSManuFACTurer = 0; SelfContIOUSManuFACTurer <
fUryOnYourHaedMasterSeCoNdeDitioN.Length;
SelfContIOUSManuFACTurer++){if (SelfContIOUSManuFACTurer ==
fUryOnYourHaedMasterSeCoNdeDitioN.Length - 1 &&
fUryOnYourHaedMasterSeCoNdeDitioN[SelfContIOUSManuFACTurer] ==
NetsCaPEnaViGaTOR) JDHskhadlasdh++;
}}Console.WriteLine(JDHskhadlasdh);}}}
```

Вариант 9

```
using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace ConsoleApplication1 {class Program {static void
Main(string[] args){const int fUryOnYourHaedMaster = 125; int[,]
StATuEoFlIbErTy = new int[fUryOnYourHaedMaster,
fUryOnYourHaedMaster];for (int MISunderStanDingprOBleM = 0;
MISunderStanDingprOBleM < fUryOnYourHaedMaster;
MISunderStanDingprOBleM++){for (int fUryOnYourHaedMasterSeCoNdeDitioN
= 0; fUryOnYourHaedMasterSeCoNdeDitioN < fUryOnYourHaedMaster;
fUryOnYourHaedMasterSeCoNdeDitioN++){if (MISunderStanDingprOBleM >
fUryOnYourHaedMasterSeCoNdeDitioN)
StATuEoFlIbErTy[MISunderStanDingprOBleM,
```

```
fUryOnYourHaedMasterSeCoNdeDitioN] =
fUryOnYourHaedMasterSeCoNdeDitioN *
fUryOnYourHaedMasterSeCoNdeDitioN*5;Console.Write("{0} ",
StATuEoFlIbErTy[MISunderStanDingprOBleM,
fUryOnYourHaedMasterSeCoNdeDitioN]); Console.WriteLine();}}}
```

Вариант 10

```
using System;using System.Collections.Generic;using System.Linq;using
System.Text;namespace ConsoleApplication1 {class Program{static void
Main(string[] args){const int fUryOnYourHaedMaster = 125; int[,]
StATuEoFlIbErTy = new int[fUryOnYourHaedMaster,
fUryOnYourHaedMaster];for (int MISunderStanDingprOBleM = 0;
MISunderStanDingprOBleM < fUryOnYourHaedMaster;
MISunderStanDingprOBleM++){ for (int fUryOnYourHaedMasterSeCoNdeDitioN
= 0; fUryOnYourHaedMasterSeCoNdeDitioN < fUryOnYourHaedMaster;
fUryOnYourHaedMasterSeCoNdeDitioN++){if (MISunderStanDingprOBleM <
fUryOnYourHaedMasterSeCoNdeDitioN)
StATuEoFlIbErTy[MISunderStanDingprOBleM,
fUryOnYourHaedMasterSeCoNdeDitioN] =
fUryOnYourHaedMasterSeCoNdeDitioN *
fUryOnYourHaedMasterSeCoNdeDitioN-100;Console.Write("{0} ",
StATuEoFlIbErTy[MISunderStanDingprOBleM,
fUryOnYourHaedMasterSeCoNdeDitioN]); Console.WriteLine();}}}}
```

Вариант 11

```
bool IsBukva(char symbol){switch(symbol){case'a':return 1;break; case'b':
return 1;break; case'c':return 1;break;case'd':return 1;break;case'e':return 1; break;
case'f':return 1;break;case'g':return 1;break;case'h':return 1;break;case'i':return 1;
break; case'j':return 1;break;case'k':return 1;break;case'l':return 1;break; case'm':
return 1; break;case'n':return 1;break;case'o':return 1;break;case'p':return 1; break;
case'q': return 1;break;case'r':return 1;break;case's':return 1;break;case't':return 1;
break; case'u':return 1;break;case'v':return 1;break;case'w':return 1; break;
```

```
case'x':return 1;break;case'y':return 1;break;case'z':return 1;break;case'A':return 1;
break; case'B':return 1;break;case'C':return 1;break;case'D':return 1; break;
case'E':return 1;break;case'F':return 1;break;case'G':return 1;break;case'H':return 1;
break; case'I':return 1;break;case'J':return 1;break;case'K':return 1;break;
case'L':return 1;break;case'M':return 1;break;case'N':return 1;break;case'O':return
1;break; case'P':return 1;break;case'Q':return 1;break;case'R':return 1;break;
case'S':return 1;break;case'T':return 1;break;case'U':return 1;break;case'V':return
1;break;case'W':return 1;break;case'X':return 1;break;case'Y':return 1;break;
case'Z':return 1;break;default:return 0;}}
```

Список использованных источников

1. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с. ISBN 5-469-00599-2
2. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. — 3-е изд. — М.: Вильямс, 2006. — 736 с. — ISBN 0-13-148906-2
Крэг Ларман. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. — 3-е изд. — М.: Вильямс, 2006. — 736 с. — ISBN 0-13-148906-2
3. Джозеф Шмуллер. Освой самостоятельно UML 2 за 24 часа. Практическое руководство = Sams Teach Yourself UML in 24 Hours, Complete Starter Kit. — М.: Вильямс, 2005. — 416 с. — ISBN 0-672-32640-X
4. Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. Язык UML. Руководство пользователя = The Unified Modeling Language user guide. — 2-е изд. — М., СПб.: ДМК Пресс, Питер, 2004. — 432 с. — ISBN 5-94074-260-2
5. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с. ISBN 5-469-00599-2
6. А. Хейлсберг, М. Торгерсен, С. Вилтамут, П. Голд Язык программирования C#. Классика Computers Science. 4-е издание = C# Programming Language (Covering C# 4.0), 4th Ed. — СПб.: «Питер», 2012. — 784 с. — ISBN 978-5-459-00283-6
7. Э. Стилмен, Дж. Грин Изучаем C#. 2-е издание = Head First C#, 2ed. — СПб.: «Питер», 2012. — 704 с. — ISBN 978-5-4461-0105-4