



МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ –  
МСХА имени К.А. ТИМИРЯЗЕВА»  
(ФГБОУ ВО РГАУ - МСХА имени К.А. Тимирязева)

Институт экономики и управления АПК  
Кафедра прикладной информатики

УТВЕРЖДАЮ:

Начальник УМУ  А.Матвеев  
“ 15 ”  2020 г.



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОХОЖДЕНИЮ ПРАКТИКИ  
Б2.О.01.02(У) УЧЕБНАЯ ОЗНАКОМИТЕЛЬНАЯ ПРАКТИКА ПО ПРО-  
ГРАММИРОВАНИЮ**

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.03 «Прикладная информатика»

Направленность: «Прикладная информатика в экономике»

Курс 2

Семестр 4

Форма обучения: очная

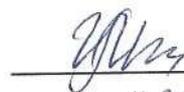
Москва, 2020

Разработчик д.э.н., проф. Худякова Е.В., Белоярская Т.С.




«25» 11 2019г.

Рецензент: Макунина И.В., к.э.н., доцент



«25» 11 2019г.

Методические указания обсуждены на заседании кафедры  
прикладной информатики «3» 12 2019 г., протокол № 4

Зав. кафедрой прикладной информатики  
Худякова Е.В., д.э.н., профессор

(ФИО, ученая степень, ученое звание)



(подпись)

**Согласовано:**

Начальник учебного  
отдела УМУ



«4» 12 2019г.

Заместитель директора по практике и профориентационной работе института  
экономики и управления АПК  
Коломеева Е.С., к.э.н., ст. преподаватель



«3» 12 2019г.

Председатель учебно-методической  
комиссии института экономики и управления АПК  
Корольков А.Ф., к.э.н., доцент



«4» 12 2019г.

**Бумажный экземпляр и копия электронного варианта получены:**  
Методический отдел УМУ

«  » \_\_\_\_\_ 20  г

## Содержание

<b>АННОТАЦИЯ .....</b>	<b>4</b>
<b>1. ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ .....</b>	<b>4</b>
<b>2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ПРАКТИКИ.....</b>	<b>4</b>
<b>3. ОРГАНИЗАЦИЯ И РУКОВОДСТВО ПРАКТИКОЙ.....</b>	<b>6</b>
3.1. Обязанности руководителя учебной практики.....	6
Обязанности студентов при прохождении учебной практики .....	7
3.2 Инструкция по технике безопасности .....	7
3.2.1. Общие требования охраны труда .....	7
<b>4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРОГРАММЫ ПРАКТИКИ .....</b>	<b>9</b>
4.1. Документы, необходимые для аттестации по практике .....	9
4.2. Общие требования к выполнению заданий по практике .....	9
<b>5. КРИТЕРИИ ОЦЕНКИ УМЕНИЙ, НАВЫКОВ (В ТОМ ЧИСЛЕ И ЗАЯВЛЕННЫХ КОМПЕТЕНЦИЙ) .....</b>	<b>17</b>
<b>6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРАКТИКИ .....</b>	<b>21</b>
6.1 Основная литература .....	21
6.2 Дополнительная литература .....	21
6.3. Программное обеспечение и Интернет-ресурсы.....	22

## АННОТАЦИЯ

### на программу практики

**Б2.О.01.02(У) «Учебная ознакомительная практика по программированию» для подготовки бакалавра по направлению 09.03.03 «Прикладная информатика», профиль «Прикладная информатика в экономике»**

**Курс, семестр:** 2 курс, 4 семестр.

**Форма проведения практики:** концентрированная, групповая.

**Способ проведения:** стационарная.

**Цель практики:** закрепить знания и навыки в области алгоритмизации и программирования на языках высокого уровня, развить способность применять естественнонаучные знания для решения задач предметной области.

**Требования к результатам прохождения практики:** в результате прохождения практики формируются следующие компетенции и индикаторы: ОПК-1.1; ОПК-7.1; ОПК-7.2; ОПК-7.3.

**Краткое содержание практики:** практика состоит из трех этапов: подготовительного, основного и заключительного.

На подготовительном этапе проводится инструктаж и обзорная лекция по теории, рассматриваемой в ходе прохождения практики.

Основной этап включает в себя выполнение и защиту практических заданий.

На заключительном этапе пишется итоговая контрольная работа по практике.

Место проведения: РГАУ-МСХА им. К.А. Тимирязева.

**Общая трудоемкость практики составляет:** 2 зач. ед. (72 часа).

**Итоговый контроль по практике:** зачет.

### 1. Цели и задачи практики

Цель прохождения учебной ознакомительной практики по программированию - закрепить знания и навыки в области алгоритмизации и программирования на языках высокого уровня, развить способность применять естественнонаучные знания для решения задач предметной области.

Задачи практики:

- изучить теоретические основы работы с графикой на языке C#;
- изучить инструментарий среды и языка для работы с графикой;
- получить умения и навыки, требуемые для разработки приложений с использованием графики.

### 2. Структура и содержание учебной практики

Студент выполняет программу практики в форме выполнения практических заданий в компьютерном классе и написания итоговой контрольной работы. Специальных документов для аттестации по программе практике не требуется.

## Содержание практики по дням прохождения

### День 1

Подготовительный этап. Инструктаж по технике безопасности, вводная лекция

Инструктаж по технике безопасности напоминает студентам, проходящим практику, о способах безопасной работы в компьютерных классах вуза.

Цель вводной лекции: познакомить студентов со структурой изучаемого материала, целями и задачами практики.

Форма контроля: по данному разделу не предусмотрен отдельный контроль, знания, полученные на лекции, проверяются в ходе выполнения практических работ в следующие дни практики.

### День 2 Основной этап

Знакомство с классами для работы с графикой

Цель: получить представление о классах для работы с графикой и навыки работы с ними.

Студентам предлагается к выполнению практическое задание, которое необходимо выполнить на ПК.

Для выполнения задания студенты могут использовать глобальные сети (непосредственно на занятиях), а также любые другие источники (в рамках самостоятельной работы). Настроенная и проверенная на работоспособность система представляется преподавателю.

Форма контроля: После выполнения задания производится индивидуальная защита выполненного задания.

### День 3- 4 Основной этап

Разработка приложения генерации ландшафта методом регулярной сетки

Цель: получить теоретические знания о методах генерации ландшафта и реализовать их на практике.

Студентам предлагается к выполнению практическое задание, которое необходимо выполнить на ПК.

Для выполнения задания студенты могут использовать глобальные сети (непосредственно на занятиях), а также любые другие источники (в рамках самостоятельной работы). Настроенная и проверенная на работоспособность система представляется преподавателю.

Форма контроля: После выполнения задания производится индивидуальная защита выполненного задания.

### День 5- 7 Основной этап

Разработка графического редактора

Цель: получить теоретические знания о методах взаимодействия пользователя с программой при реализации рисования, получить навыки разработки приложения типа «графический редактор».

Студентам предлагается к выполнению практическое задание, которое необходимо выполнить на ПК.

Для выполнения задания студенты могут использовать глобальные сети (непосредственно на занятиях), а также любые другие источники (в рамках само-

стоятельной работы). Настроенная и проверенная на работоспособность система представляется преподавателю.

Форма контроля: После выполнения задания производится индивидуальная защита выполненного задания.

### **День 8 Заключительный этап**

Обучающиеся заканчивают работу по теме «Разработка графического редактора», защищают ее и пишут контрольную работу.

Форма контроля: контрольная работа.

Форма самостоятельной работы представлена в разделе ниже.

## **3. Организация и руководство практикой**

### **3.1. Обязанности руководителя учебной практики**

#### **Назначение.**

Для руководства практикой студента, проводимой в Университете, назначается руководитель (руководители) практики из числа профессорско-преподавательского состава Университета.

**Ответственность.** Руководитель практики отвечает перед заведующим кафедрой, директором института, заместителем директора по практике и профориентационной работе института и проректором по учебно-методической работе за организацию и качественное проведение практики, и выполнение обучающимися программы практики.

Руководитель практики несет ответственность за правильное расходование средств, выделенных на проведение практики, обеспечивает соблюдение правил охраны труда и техники безопасности при проведении практики, правил трудовой и общественной дисциплины всеми практикантами.

#### ***Руководители учебной (стационарной) практики от Университета:***

- Составляет рабочий график (план) проведения практики.
- Разрабатывают тематику заданий для самостоятельной работы и оказывают методическую помощь студентам при выполнении ими индивидуальных заданий.
- Проводит инструктаж студентов по вопросам охраны труда, техники безопасности, пожарной безопасности и вопросам содержания практики проводит руководитель практики на месте её проведения с регистрацией в журнале инструктажа.
- Обеспечивает безопасные условия прохождения практики обучающимся, отвечающие санитарным правилам и требованиям охраны труда.
- Осуществляют контроль соблюдения сроков практики и её содержания.
- Распределяют студентов по рабочим местам и перемещают их по видам работ.
- Оценивают результаты выполнения студентами программы практики.
- Представляют в дирекцию института отчет о практике по вопросам, связанным с её проведением.

## ***Обязанности студентов при прохождении учебной практики***

Студенты при прохождении практики:

1. Выполняют задания (групповые и индивидуальные), предусмотренные программой практики.
2. Соблюдают правила внутреннего трудового распорядка, требования охраны труда и пожарной безопасности.
3. Оформляют учебно-методические материалы, предусмотренные программой практики, в которые записывают данные о характере и объеме практики, методах её выполнения.
4. Представляют своевременно руководителю практики письменный отчет о выполнении всех заданий, отзыв от руководителя практики от Организации и сдают зачет по практике в соответствии с формой аттестации результатов практики, установленной учебным планом с учетом требований ФГОС ВО и ОПОП.
5. Несут ответственность за выполняемую работу и её результаты.
6. При неявке на практику (или часть практики) по уважительным причинам обучающиеся обязаны поставить об этом в известность институт и в первый день явки в университет представить данные о причине пропуска практики (или части практики). В случае болезни обучающийся представляет в дирекцию института справку установленного образца соответствующего лечебного учреждения.

### ***3.2 Инструкция по технике безопасности***

Перед началом практики заместители заместители директоров по практике и профориентационной работе и руководители практики от Университета проводят инструктаж студентов по вопросам охраны труда, техники безопасности, пожарной безопасности, производственной санитарии и общим вопросам содержания практики с регистрацией в журнале инструктажа и вопросам содержания практики.

#### ***3.2.1. Общие требования охраны труда***

К самостоятельной работе допускаются лица в возрасте, установленном для конкретной профессии (вида работ) ТК и Списком производств, профессий и работ с тяжелыми и вредными условиями труда, на которых запрещается применение труда женщин, и Списком производств, профессий и работ с тяжелыми и вредными условиями, на которых запрещено применение труда лиц моложе 18 лет.

Обучающиеся должны проходить предварительный медицинский осмотр и, при необходимости, периодический осмотр и противоэнцефалитные прививки. После этого – обучение по охране труда: вводный инструктаж, первичный на рабочем месте с последующей стажировкой и в дальнейшем – повторный, внеплановый и целевой инструктажи; раз в год – курсовое обучение.

К управлению машиной, механизмом и т.д. допускаются лица, имеющие специальную подготовку.

Обучающийся обязан соблюдать правила трудового внутреннего распорядка, установленные для конкретной профессии и вида работ, режим труда и отдыха, правила пожарной и электробезопасности.

Опасные и вредные производственные факторы: падающие деревья и их части, ветровально-буреломные, горелые, сухостойные, фаутные и иные опасные деревья, подрост, кустарники; движущиеся машины, агрегаты, ручной мотоинструмент, вращающиеся части и режущие рабочие органы машин, механизмов, мотоинструмента, толчковые удары лесохозяйственных агрегатов; повышенные уровни вибрации, шума, загазованности, запыленности, пестициды и ядохимикаты, неблагоприятные природные и метеоусловия, кровососущие насекомые, пламя, задымленность, повышенный уровень радиации, недостаток освещенности.

Действие неблагоприятных факторов: возможность травмирования и получения общего или профессионального заболевания, недомогания, снижение работоспособности.

Для снижения воздействия на обучающихся опасных и вредных производственных факторов работодатель обязан: обеспечить их бесплатно спецодеждой, спецобувью, предохранительными приспособлениями по профессиям, видам работ в соответствии с действующими Типовыми отраслевыми нормами бесплатной их выдачи и заключенными коллективными договорами, проведение прививок от клещевого энцефалита и иных профилактических мероприятий травматизма и заболеваемости.

Обучающийся обязан: выполнять работу, по которой обучен и проинструктирован по охране труда и на выполнение которой он имеет задание; выполнять требования инструкции по охране труда, правила трудового внутреннего распорядка, не распивать спиртные напитки, курить в отведенных местах и соблюдать требования пожарной безопасности; работать в спецодежде и обуви, правильно использовать средства индивидуальной и групповой защиты, знать и соблюдать правила проезда в пассажирском транспорте.

При несчастном случае необходимо: оказать пострадавшему первую помощь (каждый обучающийся должен знать порядок ее оказания и назначение лекарственных препаратов индивидуальной аптечки); по возможности сохранить обстановку случая, при необходимости вызвать скорую помощь и о случившемся доложить непосредственному руководителю работ.

Обо всех неисправностях работы механизмов, оборудования, нарушениях технологических режимов, ухудшении условий труда, возникновении чрезвычайных ситуаций сообщить администрации и принять профилактические меры по обстоятельствам, обеспечив собственную безопасность.

В соответствии с действующим законодательством обучающийся обязан выполнять требования инструкций, правил по охране труда, постоянно и правильно использовать средства индивидуальной и групповой защиты. Своевременно проходить предварительные и периодические медицинские осмотры, противэнцефалитные прививки и иные меры профилактики заболеваемости и травматизма.

## 4. Методические указания по выполнению программы практики

### 4.1. Документы, необходимые для аттестации по практике

Специальных документов для аттестации по программе практике не требуется.

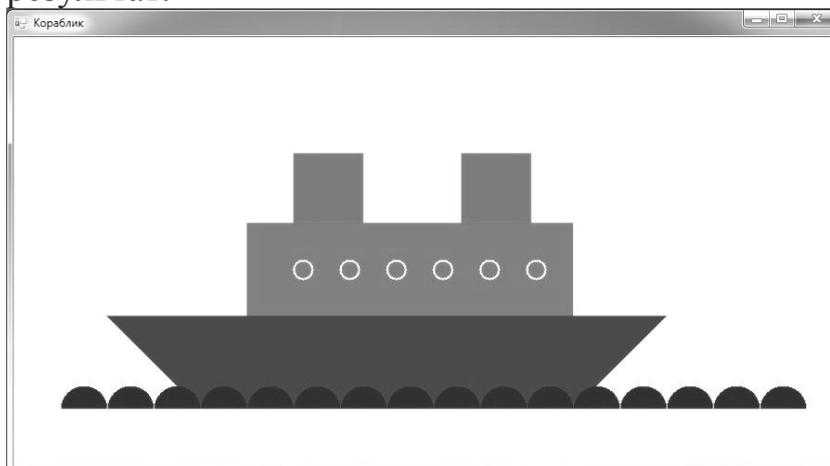
### 4.2. Общие требования к выполнению заданий по практике

#### Задания по практике

##### Знакомство с классами для работы с графикой

Цель работы: познакомиться с классами и их свойствами и методами, используемыми в C# для рисования.

Примерный результат:



#### Ход работы:

- 1) Создайте приложение Windows Forms.
- 2) Размер формы задайте 900x500, на ней разместите визуальный объект для рисования `pictureBox1` (имя – `PBShip`), измените его свойство `Dock = Fill`; (он займет все пространство формы).
- 3) Не забудьте добавить библиотеку `System.Drawing`, если ее нет в области `using`.
- 4) Создайте обработчик щелчка по рисунку.
- 5) В коде обработчика будем осуществлять рисование.
  - 5.1 Для начала создайте объект "g" класса `Graphics`. Это делается через присваивание переменной значения из `PBShip.CreateGraphics()`.
  - 5.2 Далее очистим поле картинку заливкой - `g.Clear(Color.Turquoise)`.
  - 5.3 Создаем объекты-кисти для закрашивания фигур:  
`SolidBrush myCorp = new SolidBrush(Color.DarkMagenta);`  
(`myTrum`, `myTrub`, `mySea` – создайте эти переменные по аналогии)  
Выбираем перо `myPen` желтого цвета толщиной в 2 пикселя:  
`Pen myWind = new Pen(Color.Yellow, 2);`
  - 5.4 Рисуем и закрашиваем фигуры:  
`g.FillRectangle(myTrub, 300, 125, 75, 75);` // 1 труба (прямоугольник)  
(вторую трубу и трюм дорисуйте по аналогии)  
Корпус рисуется с помощью полигона  
`g.FillPolygon(myCorp, masPoint)` // корпус (трапеция)

(masPoint – массив точек полигона, создайте его перед рисованием)

More - 12 секторов-полуокружностей

int x = 50;

int Radius = 50;

Далее в цикле до PBShip.Width – Radius необходимо чертить сектора окружности - g.FillPie(mySea, 0 + x, 375, 50, 50, 0, -180); с шагом 50 пикселей

Иллюминаторы – это круги, которые мы чертим в цикле от 300 до 550 с шагом 50

g.DrawEllipse(myWind, y, 240, 20, 20); // 6 окружностей

б) Проверить работу приложения и сдать ее преподавателю.

### **Разработка приложения генерации ландшафта методом регулярной сетки**

**Постановка задачи:** разработать приложение, реализующее генерацию ландшафта методом регулярной сетки высот со случайными значениями.

#### **Метод регулярной сетки**

Данные представлены в виде двумерного массива (рис. 1). Уже заданы две координаты (x, y — по высоте и ширине массива), и третья координата задается значением в конкретной ячейке, это высота.

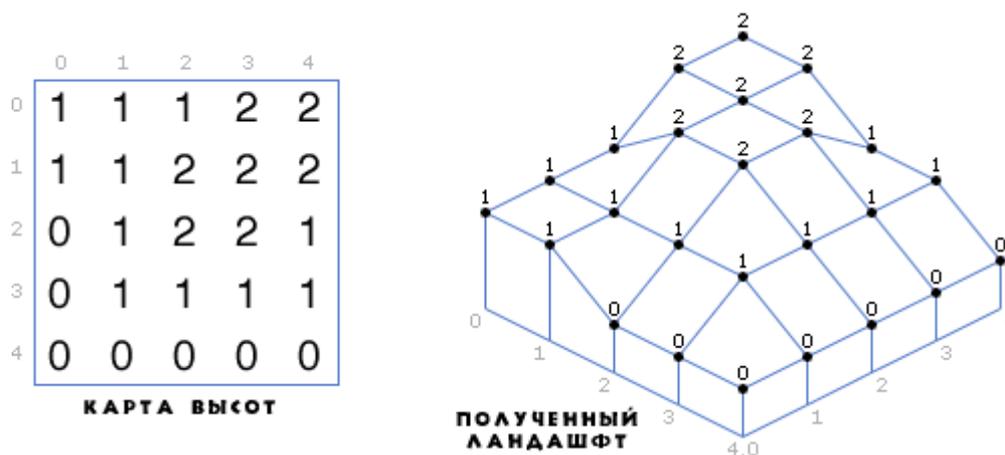


Рис. 1 Пример создания ландшафта на основе регулярной сетки (карты высот)

Обычно карту высот хранят в файлах картинок. Это позволяет легко вносить изменения и более-менее наглядно просматривать данные. Тогда двумя координатами будет определено положение конкретного пикселя на картинке, а третья координата будет представлена цветом (чем выше значение, прямая зависимость от яркости пикселя — тем больше значение высоты для этой точки). Обычно такие картинки содержатся в монохромном варианте, но можно использовать и все цвета радуги. Второй вариант дает нам больше градаций высоты, чем предполагаемые 256 градаций в случае монохромного представления.

#### **Ход работы:**

1. Создать приложение с графическим интерфейсом и сохранить в отдельную папку.
2. Нанести на форму компоненты – PictureBox, 3 кнопки Button.

3. Переименовать компоненты: форму – Main, картинку - Img1, 1 кнопку – btnGen, 2 кнопку – btnAvg, 3 кнопку – btnClose.
4. Подписи для формы и кнопок, как на рис. 2.
5. Установить размеры изображения – 300x300.

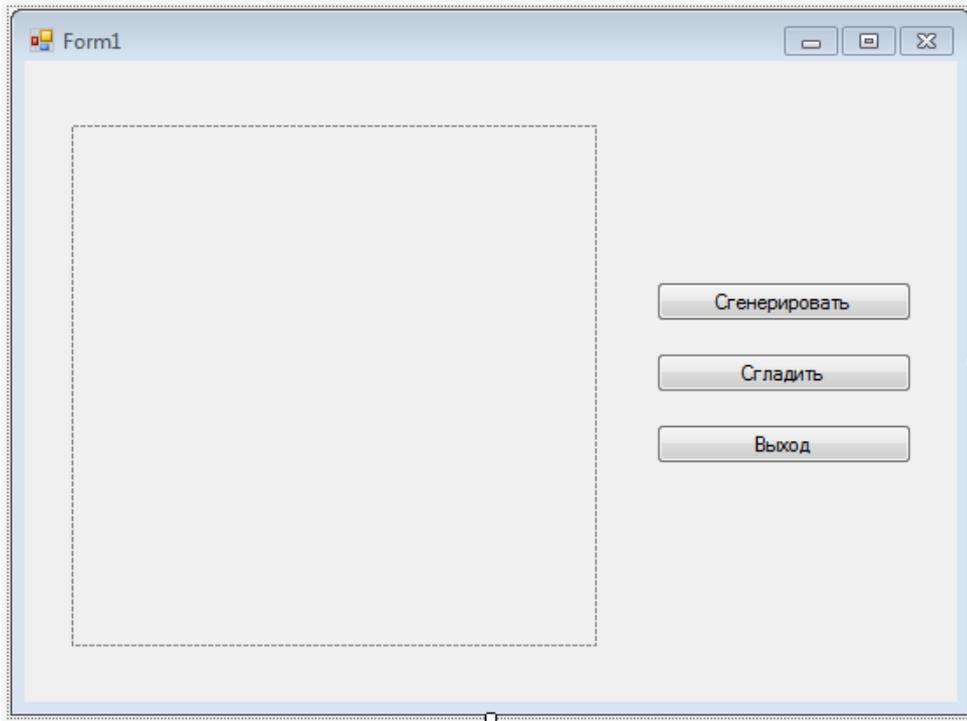


Рис. 2 Примерный интерфейс приложения

6. Объявить 2 глобальных массива ArData и AvgData. Первый для хранения случайных значений величин, второй – для усредненных значений.

```
float[,] ardata = new float[300, 300];
float[,] avgdata = new float[300, 300];
```

7. Создайте процедуры для обработки нажатия на кнопку «Сгенерировать».

8. Переопределите массив ArData – SetLength(ArData,300,300) и заполните его случайными вещественными числами от 0 до 100.

```
private void btnGen_Click(object sender, EventArgs e)
{
    Random random = new Random();
    for (i = 0; i < 300; i++)
    {
        for (j = 0; j < 300; j++)
        {
            ardata[i, j] = random.Next(100) ;
        }
    }
}
```

9. Создайте процедуру SetPixel\_Example. В данной процедуре на точечном рисунке будут генерироваться пиксели определенных цветов в соответствии со значениями массива.

```
Bitmap pic = new Bitmap(Img1.Width, Img1.Height);
```

10. Например, от 0 до 33 – красные, от 33 до 66 – зеленые, от 66 до 99 – синие. В дальнейшем можно будет использовать свои цвета и градации.

11. Разработайте цветовую легенду с указанием высот (рис. 2, можно использовать свои цвета и градации).

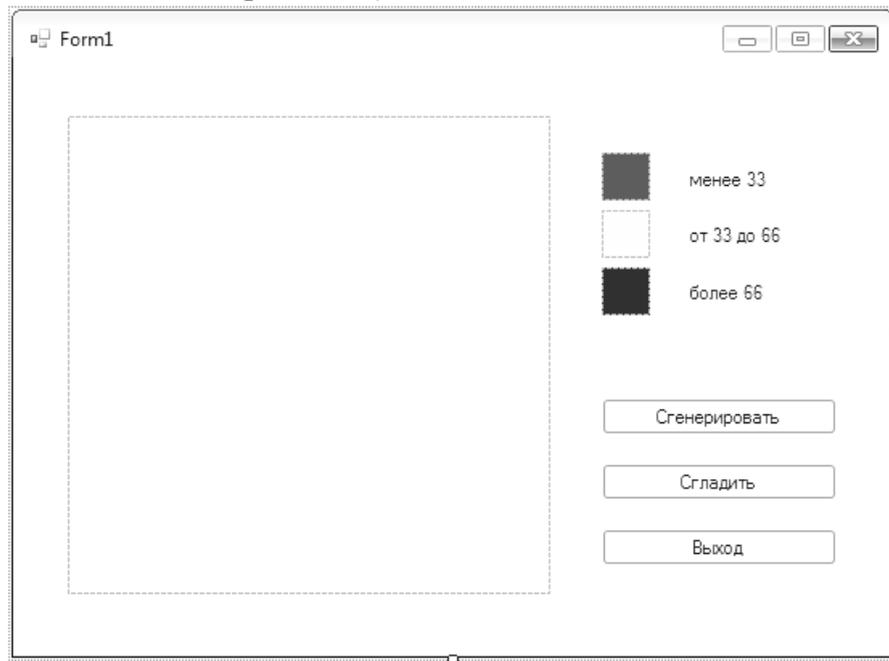


Рис. 2 Легенда для цветовой градации высот

12. Для создания легенды можно использовать три компонента PictureBox размером 30x30 и загрузить в них файлы \*.bmp, соответствующего размера и цвета.

13. Теперь запишем код в процедуру SetPixel\_Example.

Для двумерных массивов вся работа проходит во вложенных циклах –

```
for (int Xcount = 0; Xcount < pic.Width; Xcount++)
```

```
{
```

```
    for (int Ycount = 0; Ycount < pic.Height;
```

```
Ycount++)
```

Необходимо проверить каждое значение массива на принадлежность одной из цветовых категорий и покрасить соответствующий значению пиксель изображения.

```
{
    if (ardata[Xcount, Ycount] < 33 )
pic.SetPixel(Xcount, Ycount, Color.Red);
    if ((ardata[Xcount, Ycount] < 66) && (ardata[Xcount,
Ycount] > 33))
pic.SetPixel(Xcount, Ycount, Color.Green);
    if ((ardata[Xcount, Ycount] < 100) && (ardata[Xcount,
Ycount] > 66))
pic.SetPixel(Xcount, Ycount, Color.Blue);
}
```

14. В конец процедуры нажатия кнопки добавить вызов процедуры XetPixel\_Example:

```
SetPixel_Example(null);
```

15. Проверить работоспособность кода.

16. Создать процедуру для обработки нажатия кнопки «Сгладить».

17. Переопределите массив AvgData так же как основной массив. Рассчитайте его элементы по формуле среднего арифметического. Всего используется 9 формул.

```
for (i = 0; i < 300; i++)
{
    for (j = 0; j < 300; j++)
    {
        if ((i == 0) && (j == 0))
            { avgdata[i, j] = (ardata[i, j] +
ardata[i + 1, j] + ardata[i + 1, j + 1] + ardata[i, j +
1]) / 4; ;
        }
    }
}
```

18. Создайте процедуру SetPixel\_Example2, которая будет отрисовывать пиксели в соответствии со значениями массива AvgData.

```
private void SetPixel_Example2(PaintEventArgs e)
{
    Bitmap pic = new Bitmap(Img1.Width, Img1.Height);
    for (int Xcount = 0; Xcount < pic.Width; Xcount++)
    {
        for (int Ycount = 0; Ycount < pic.Height;
Ycount++)
        {
            if (avgdata[Xcount, Ycount] < 33)
                pic.SetPixel(Xcount, Ycount, Color.Red);
            if ((avgdata[Xcount, Ycount] < 66) &&
(avgdata[Xcount, Ycount] > 33))
                pic.SetPixel(Xcount, Ycount, Color.Green);
            if ((avgdata[Xcount, Ycount] < 100) &&
(avgdata[Xcount, Ycount] > 66))
                pic.SetPixel(Xcount, Ycount, Color.Blue);
        }
    }
    Img1.Image = pic;
}
```

19. Проверить работоспособность программы.

### ***Разработка графического редактора***

**Постановка задачи:** Разработать графический редактор, осуществляющий рисование простейших фигур: линия, прямоугольник, круг, позволяет задать заливку, границу фигур и ее толщину, а также предусматривает возможность рисования по типу карандаша.

#### **Ход работы:**

1. Создать новый проект с графическим интерфейсом.

2. На форме разместить компоненты PictureBox, SaveFileDialog и OpenFileDialog.

3. Компонент PictureBox переименовать в Img1.

4. Добавить компонент ToolStripMenuItem, переименовать его в Menu1.

5. Добавить туда два пункта – «Сохранить» и «Загрузить».

6. Добавить пять кнопок. Каждая кнопка будет обозначать инструменты графического редактора: линия, карандаш, ластик, прямоугольник, эллипс. Переименовать кнопки в btnLine, btnPen, btnEraser, btnRect, btnEll соответственно. Удалить текст из кнопок (очистить свойство Text) в свойство BackgroundImage добавить ссылку на соответствующую картинку. Для того чтобы растянуть картинку по всей кнопке, необходимо раскрыть свойство BackgroundImage и в появившемся свойстве BackgroundImageLayout выбрать опцию Stretch.

7. Для учета включенного инструмента (линия, карандаш, ластик, прямоугольник, эллипс) объявить глобальные переменные типа Boolean – bLine, bPen, bEraser, bRect, bEll.

8. Также для включения режима рисования необходимо объявить глобальную переменную типа Boolean под названием IsClicked.

9. Объявить четыре глобальные переменные типа integer и присвоить им значение 0:

```
int X = 0;  
int X1 = 0;  
int Y = 0;  
int Y1 = 0;
```

10. Создать обработчики события нажатия для всех кнопок и прописать код включения соответствующего режима и отключения остальных (пример для кнопки btnLine)

```
bLine:=True;  
bPen:=False;  
bEraser:=False;  
bRect:=False;  
bEll:=False;
```

11. В процедуре создания формы отключить все режимы, присвоив логическим переменным значения false.

12. Для реализации работы приложения создать обработчики событий MouseDown, MouseUp, MouseMove компонента Img1.

13. В событии img1\_MouseDown присвоить переменной IsClicked значение true и записать положение курсора в созданные ранее переменные:

```
X = e.X;  
Y = e.Y;
```

14. Реализуем рисование линии в методе Img1\_Paint.

Сначала создадим экземпляр графического интерфейса dLine, на котором будут отрисовываться фигуры:

```
Graphics dline = e.Graphics;
```

А так же экземпляр пера dPen, в котором задаётся цвет и толщина линии

```
Pen dPen = new Pen(Color.Black, 2);
```

Далее при выполнении условия `bLine == true` необходимо прорисовать линию с помощью функции `DrawLine`:

```
dLine.DrawLine(dPen, new Point(X, Y), new Point(X1, Y1));
```

где `Point(X, Y)` – координаты начального положения курсора, а `Point(X1, Y1)` – координаты конечного положения курсора.

15. Конечное положение курсора при рисовании линии необходимо обозначить в методе `Img1_MouseMove`:

```
if (IsClicked == true)
{
    if ((bLine == true)
        {
            X1 = e.X;
            Y1 = e.Y;
            img1.Invalidate();
        }
}
```

16. Далее реализуем рисование прямоугольника – при нажатии левой кнопки мыши запоминаются координаты левого верхнего угла будущего прямоугольника, если отпустить зажатую кнопку, то рисуется прямоугольник – левый верхний угол – координаты начального положения курсора, правый нижний угол – координаты конечного положения курсора.

Для этого в методе `Img1_Paint` необходимо создать экземпляр прямоугольника и создать его с помощью функции `DrawRectangle`, которая в качестве параметров использует экземпляр пера `dPen` и экземпляр прямоугольника `rect` с заданными координатами:

```
if (bRect == true)
{
    Rectangle rect = new Rectangle(X, Y, X1, Y1);
    dline.DrawRectangle(dPen, rect);
}
```

17. Конечные координаты положения курсора можно получить тем же способом, что использовался в пункте 15, только для этого необходимо, чтобы соблюдалось условие `bRect == true`.

18. Рисование эллипса почти не отличается от рисования прямоугольника. Сначала создаётся экземпляр прямоугольника, в который будет вписан эллипс, а дальше идёт создание эллипса с помощью функции `DrawEllipse`.

```
if (bEll == true)
{
    Rectangle rect = new Rectangle(X, Y, X1, Y1);
    dline.DrawEllipse(dPen, rect);
}
```

19. Конечные координаты положения курсора записываются ранее использованным способом. Только для создания эллипса будет использоваться условие `bEll == true`.

20. Для создания инструментов «карандаш» и «ластик» необходимо глобально объявить экземпляры класса `SolidBrush: dBrush` и `dEraser`; а также экземпляр графического интерфейса `myGraphics`.

```
SolidBrush dBrush;  
SolidBrush dEraser;  
Graphics myGraphics;
```

21. Далее в методе `Form1_Load` необходимо задать цвет карандаша и ластика:

```
dBrush = new SolidBrush(Color.Black);  
dEraser = new SolidBrush(Color.White);
```

22. В методе `Img1_MouseMove` необходимо создать графический интерфейс в объявленном ранее экземпляре `myGraphics`. Точки, созданные при использовании карандаша или ластика в данном случае представляют собой эллипсы 5 на 5 пикселей, использующие кисти `dBrush` и `dEraser` соответственно. Конечный вид метода `Img1_MouseMove` представлен ниже:

```
private void Img1_MouseMove(object sender,  
MouseEventArgs e)  
{  
    if (IsClicked == true)  
    {  
        myGraphics = Img1.CreateGraphics();  
        if (bPen == true)  
        {  
            myGraphics.FillEllipse(dBrush, e.X,  
e.Y, 5, 5);  
        }  
        if (bEraser == true)  
        {  
            myGraphics.FillEllipse(dEraser, e.X,  
e.Y, 5, 5);  
        }  
        if ((bLine == true) || (bRect == true) ||  
(bEll== true))  
        {  
            X1 = e.X;  
            Y1 = e.Y;  
            img1.Invalidate();  
        }  
    }  
}
```

23. При запуске приложения можно заметить, что при переключении инструмента рисования созданный ранее рисунок пропадает. Это связано с тем, что создаваемый рисунок не сохраняется в памяти компьютера. Для устранения данного недостатка необходимо доработать приложение.

24. В методе `Img1_MouseUp` при активации любого из инструментов значение переменной `IsClicked` меняется на `false`.

```
if ((bPen == true) || (bEraser == true) || (bLine == true) || (bRect == true) || (bEll == true))
{
    IsClicked = false;
}
```

25. Далее необходимо создать:

- Экземпляр прямоугольника, имеющего размеры компонента `Img1`  
`Rectangle r = img1.RectangleToScreen`  
`(img1.ClientRectangle) ;`

- Экземпляр точечного рисунка `Bitmap`, размер которого равен размерам созданного ранее прямоугольника

```
Bitmap b = new Bitmap(r.Width, r.Height);
```

- Экземпляр графического интерфейса, который в качестве загружаемого рисунка использует созданный выше точечный рисунок.

```
Graphics g = Graphics.FromImage(b);
```

26. Заполнение рисунка осуществляется с помощью функции `CopyFromScreen`, которая имеет три параметра. Первый параметр вычисляет положение компонента, который будет использоваться в качестве области для копирования, второй обозначает начальные координаты копируемой области экрана, а третий – конечные координаты.

```
g.CopyFromScreen(r.Location, new Point(0, 0), r.Size);
```

27. Сохраним полученный точечный рисунок.

```
b.Save(filepath);
```

где `filepath` – полное имя сохраняемого точечного рисунка. Для удобства рекомендуется сохранить рисунок в папке с проектом.

28. Сохранение точечного рисунка через меню осуществляется аналогичным методом, только после копирования изображения с экрана вызывается диалог сохранения файла и полное имя файла задается через компонент `saveFileDialog1`:

```
saveFileDialog1.ShowDialog();
```

```
if (saveFileDialog1.FileName != "") //при несоблюдении данного условия закрытие диалога без выбора файла вызовет исключение и приложение прекратит работу
```

```
b.Save(saveFileDialog1.FileName);
```

29. Проверить работоспособность кода и сдать задание преподавателю. Самостоятельно добавить выбор цвета границ и фона, а также толщины границы.

## **5. Критерии оценки умений, навыков (в том числе и заявленных компетенций)**

### **5.1. Текущая аттестация по разделам практики**

Текущая аттестация производится путем защиты выполненной практической работы.

### **Критерии оценки практических заданий**

Задание выполняется в классе и сдается преподавателю. Оценка производится по системе зачет/не зачет. Зачет выставляется студенту в случае, если он полностью выполнил все пункты практического задания.

### **Контрольные вопросы для текущей аттестации по учебной практике**

1. Пространство имен System.Drawing
2. Класс Graphics: обзор свойств и методов.
3. Методы рисования класса Graphics: методы DrawLine(Pen, Point, Point), DrawRectangle(Pen, Rectangle).
4. Методы рисования класса Graphics: метод DrawImage().
5. Класс Pen: обзор свойств и методов.
6. Класс Brush: обзор свойств и методов.
7. Класс Bitmap: обзор свойств и методов.
8. Класс Bitmap: метод SetPixel (int x, int y, System.Drawing.Color color).
9. Компонент pictureBox: обзор свойств и методов.
10. Компонент pictureBox: свойство Image. Назначение картинки компоненту.

### **Критерии оценки:**

Вопросы текущей аттестации задаются студенту в ходе выполнения практических заданий в момент их сдачи преподавателю. Система оценивания – зачет/не зачет. Зачет выставляется в случае выполнения всех пунктов практического задания и ответа на дополнительные вопросы (не более 3-х вопросов).

## **5.2. Промежуточная аттестация по практике**

### **Контрольные задания для промежуточной аттестации**

Для получения зачета обучающийся должен написать контрольную работу.

#### **Вариант 1**

Создать приложение, визуализирующее поступление товаров на склад:

- имеется 2 отдела разной температуры хранения с заданным количеством полок (емкость полок как и их количество установлено программно);
- размеры товара определяются объемом;
- при поступлении указывается температурный режим товара;
- программа должна устанавливать полную загруженность полки;
- после визуализации передвинуть товар нельзя.
- реализовать сохранение схемы.

#### **Вариант 2**

Создать приложение для построения графика прямой по заданным пользователем параметрам (уравнение  $y=ax+b$ ). Отобразить оси координат и разметку осей.

### Вариант 3

Создать приложение, визуализирующее сроки годности товаров:

- товары заносятся в список;
- каждый товар характеризуется датой изготовления и сроком годности;
- назначить цветовую дифференциацию для степени просроченности (подошел срок, вышел срок и совсем просрочено, например);
- можно визуализировать ограниченное количество товаров;
- реализовать сохранение схемы.

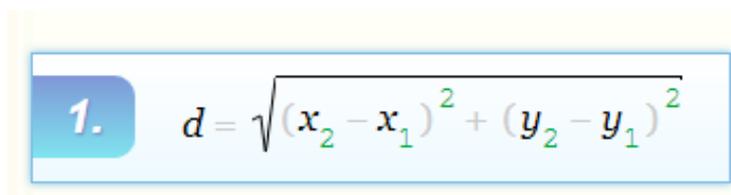
### Вариант 4

Создать приложение для построения поверхности ландшафта с параметрами пользователя:

- количество цветов ландшафта (максимальное количество цветов можно зафиксировать);
- минимальная и максимальная высота.

### Вариант 5

Создать приложение для построения поверхности ландшафта с добавлением точек и рисованием дорог по этим точкам. Для дорог рассчитать их длину по формуле:



1.  $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

**Критерии оценки:** Максимальное число баллов за контрольную работу – 25 баллов рейтинга. Балльная система оценки контрольной работы представлена в таблице 1.

Таблица 1

#### Критерии снижения баллов

Баллы	Описание
-25	Программа не запускается или же запускается, но не реализует указанный в задании функционал.
-3	Программа запускается и реализует функционал задания, но: 1) содержит лишние конструкции, не влияющие на ход выполнения программы; 2) отсутствует проверка ввода

Для перевода баллов рейтинга в традиционную систему оценок можно использовать таблицу 2.

**Перевод баллов в традиционную систему оценивания**

Баллы	Балльная оценка текущей успеваемости	
За контрольную работу	0-14	15-25
Оценка	Не зачтено	Зачтено

**Вопросы промежуточной аттестации (зачета):**

1. Пространство имен System.Drawing.
2. Класс Graphics: обзор свойств и методов.
3. Методы рисования класса Graphics: методы DrawLine(Pen, Point, Point), DrawRectangle(Pen, Rectangle).
4. Методы рисования класса Graphics: метод DrawImage().
5. Класс Pen: обзор свойств и методов.
6. Класс Brush: обзор свойств и методов.
7. Класс Bitmap: обзор свойств и методов.
8. Класс Bitmap: метод SetPixel (int x, int y, System.Drawing.Color color).
9. Компонент pictureBox: обзор свойств и методов.
10. Компонент pictureBox: свойство Image. Назначение картинки компоненту.

**Вопросы практического характера:**

11. В каких случаях можно не создавать объект класса Bitmap?
12. Что представляет из себя тип Point?
13. Какие параметры задаются для метода DrawRectangle()?
14. Где в компоненте pictureBox находится координата 0,0?
15. В каких случаях можно не задавать объект Pen?
16. За счет чего можно привязать объект Graphics к компоненту pictureBox?
17. Как на этапе разработки настроить цвет pictureBox?
18. Как осуществить закраску отдельного пикселя?
19. Чем попиксельная закраска отличается от заливки?
20. Как рассчитать координаты на плоскости для pictureBox?

**Критерии оценивания результатов обучения**

Оценка	Критерии оценивания
Зачтено	Студент успешно прошел текущую аттестацию и достаточно ответил на вопросы промежуточной аттестации.
Не зачтено	Студент не прошел текущую аттестацию и/или не ответил/не полностью ответил на вопросы промежуточного контроля

Студенты, не выполнившие программы практик по уважительной причине, направляются на практику вторично, в свободное от учебы время, либо практика переносится на следующий год с оформлением соответствующего приказа.

Студенты, не выполнившие программы практик без уважительной причины, или получившие отрицательную оценку отчисляются из Университета как имеющие академическую задолженность в порядке, предусмотренном Уставом Университета.

## **6. Учебно-методическое и информационное обеспечение практики**

### **6.1 Основная литература**

1. Барков, И.А. Объектно-ориентированное программирование : учебник / И.А. Барков. — Санкт-Петербург : Лань, 2019. — 700 с. — ISBN 978-5-8114-3586-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/119661> (дата обращения: 18.02.2020). — Режим доступа: для авториз. пользователей.
2. Залогова, Л.А. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие / Л.А. Залогова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2020. — 192 с. — ISBN 978-5-8114-4757-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/126160> (дата обращения: 18.02.2020). — Режим доступа: для авториз. пользователей.

### **6.2 Дополнительная литература**

1. Симонова, Е.В. Структуры данных в C#: линейные и нелинейные динамические структуры : учебное пособие / Е.В. Симонова. — Санкт-Петербург : Лань, 2018. — 152 с. — ISBN 978-5-8114-3098-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/110938> (дата обращения: 18.02.2020). — Режим доступа: для авториз. пользователей.
2. Старолетов, С.М. Основы тестирования и верификации программного обеспечения : учебное пособие / С.М. Старолетов. — Санкт-Петербург : Лань, 2018. — 344 с. — ISBN 978-5-8114-3041-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/110939> (дата обращения: 18.02.2020). — Режим доступа: для авториз. пользователей.
3. Тюкачев, Н.А. C#. Основы программирования : учебное пособие / Н.А. Тюкачев, В.Г. Хлебостроев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2018. — 272 с. — ISBN 978-5-8114-2567-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/104962> (дата обращения: 18.02.2020). — Режим доступа: для авториз. пользователей.
4. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование : учебное пособие / С. В. Белугина. — Санкт-Петербург : Лань, 2020. — 312 с. — ISBN 978-5-8114-4496-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/133920> (дата обращения: 18.02.2020). — Режим доступа: для авториз. пользователей.

### **6.3. Программное обеспечение и Интернет-ресурсы**

Обязательное ПО:

1. ОС Windows 7, Microsoft 2009;
2. Браузер: Internet Explorer не ниже 11 версии, Google Chrome, Яндекс.Браузер или аналоги.

Дополнительное ПО:

1. Visual Studio 2010, Microsoft 2010

Интернет-ресурсы:

1. <http://www.ixbt.com> – Новости технологий, обзоры гаджетов, смартфонов, бытовой техники и автомобилей – открытый доступ;

2. <https://habr.com/ru> – русскоязычный веб-сайт в формате коллективного блога с элементами новостного сайта, созданный для публикации новостей, аналитических статей, мыслей, связанных с информационными технологиями, бизнесом и интернетом– открытый доступ.