

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Наумова Наталья Александровна
Должность: Ректор
Дата подписания: 24.10.2024 14:21:41
Уникальный программный ключ:
6b5279da4e034bffa679172803da5b7b559fc69e2

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПРОСВЕЩЕНИЯ»
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПРОСВЕЩЕНИЯ)

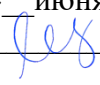
Физико-математический факультет

Кафедра вычислительной математики и информационных технологий

УТВЕРЖДЕН

на заседании кафедры

Протокол от «_10_» июня 2024 г., №_15_

Зав. кафедрой  [Шевчук М.В.]

**ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине (модулю)

Программирование в визуальных средах

Направление подготовки (специальности) 44.04.01 Педагогическое образование
Профиль (программа подготовки, специализация) Современные информационные
образовательные технологии

Мытищи
2024

Содержание

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	3
2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ.....	3
3. КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	7
4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ.....	37

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Код и наименование компетенции	Этапы формирования
УК-2. Способен управлять проектом на всех этапах его жизненного цикла	1. Работа на учебных занятиях 2. Самостоятельная работа
СПК-2 «Способен к преподаванию учебных курсов, дисциплин (модулей) по образовательным программам в образовательных организациях соответствующего уровня образования»	1. Работа на учебных занятиях 2. Самостоятельная работа

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Оцениваемые компетенции	Уровень сформированности	Этап формирования	Описание показателей	Критерии оценивания	Шкала Оценивания
УК-2	Пороговый	1. Работа на учебных занятиях Самостоятельная работа	Знать: - теоретические основы проектов в рамках жизненного цикла; - методологию проектной деятельности; - современные концепции, теории, законы и методы в области информатики; - основные понятия и сопряженные с ними из предметной области; - систему диагностики и оценки реализованного проекта; Уметь: - планировать и осуществлять все этапы проекта в рамках жизненного цикла; - использовать систему диагностики и оценки реализованного проекта; - профессионально решать задачи, связанные с предметной областью, с учетом современных достижений науки;	Лабораторная работа Самостоятельная работа	Шкала оценивания лабораторной работы Шкала оценивания самостоятельной работы

	Продвинутый	1. Работа на учебных занятиях Самостоятельная работа	<i>Знать:</i> - теоретические основы проектов в рамках жизненного цикла; - методологию проектной деятельности; - современные концепции, теории, законы и методы в области информатики; - основные понятия и сопряженные с ними из предметной области; - систему диагностики и оценки реализованного проекта; <i>Уметь:</i> - планировать и осуществлять все этапы проекта в рамках жизненного цикла; - использовать систему диагностики и оценки реализованного проекта; - профессионально решать задачи, связанные с предметной областью, с учетом современных достижений науки; <i>Владеть:</i> - опытом (навыками) использования системы диагностики и оценки реализованного проекта; - опытом планирования реализации проекта в соответствующей предметной области.	Лабораторная работа Самостоятельная работа	Шкала оценивания лабораторной работы Шкала оценивания самостоятельной работы
СПК-2	Пороговый	1. Работа на учебных занятиях 2. Самостоятельная работа	<i>Знать:</i> - теоретические основы программирования; - методологию, теорию и эффективную практику образовательной деятельности; - современные концепции, теории, законы и методы в	Лабораторная работа Самостоятельная работа	Шкала оценивания лабораторной работы Шкала оценивания самостоятельной работы

			области информатики и перспективные направления развития современной науки; -особенности планирования и осуществления учебного процесса в соответствующей предметной области; -основные понятия и сопряженные с ними из предметной области; - систему диагностики и оценки уровня образовательных достижений обучающихся; <i>Уметь:</i> - планировать и осуществлять все составляющие учебного процесса в соответствии в соответствующей предметной области; - использовать систему диагностики и оценки уровня образовательных достижений обучающихся; - профессионально решать задачи, связанные с предметной областью, с учетом современных достижений науки;		
	Продвинутый	1. Работа на учебных занятиях 2. Самостоятельная работа	<i>Знать:</i> - теоретические основы программирования; - методологию, теорию и эффективную практику образовательной деятельности; - современные концепции, теории, законы и методы в области информатики и перспективные направления развития современной науки;	Лабораторная работа Самостоятельная работа	Шкала оценивания лабораторной работы Шкала оценивания самостоятельной работы

			-особенности планирования и осуществления учебного процесса в соответствующей предметной области; -основные понятия и сопряженные с ними из предметной области; - систему диагностики и оценки уровня образовательных достижений обучающихся; <i>Уметь:</i> - планировать и осуществлять все составляющие учебного процесса в соответствии в соответствующей предметной области; - использовать систему диагностики и оценки уровня образовательных достижений обучающихся; - профессионально решать задачи, связанные с предметной областью, с учетом современных достижений науки; <i>Владеть:</i> - опытом (навыками) использования системы диагностики и оценки уровня образовательных достижений обучающихся; - опытом планирования учебного процесса в соответствующей предметной области.		
--	--	--	--	--	--

Шкала оценивания лабораторных работ

Критерии оценивания	Баллы
---------------------	-------

Задание выполнено частично; в оформлении отчета есть существенные неточности или оформление отчета вообще не соответствует заявленным требованиям	4
Задание выполнено полностью, но в оформлении отчета есть некоторые неточности	6
Задание выполнено полностью, оформление отчета полностью соответствует заявленным требованиям	10

Шкала оценивания самостоятельных работ

Критерии оценивания	Баллы
Задание выполнено частично; в оформлении отчета есть существенные неточности или оформление отчета вообще не соответствует заявленным требованиям	2
Задание выполнено полностью, но в оформлении отчета есть некоторые неточности	6
Задание выполнено полностью, оформление отчета полностью соответствует заявленным требованиям	8

3. Контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Текущая аттестация

УК-2 «Способен управлять проектом на всех этапах его жизненного цикла»

Знать:

- теоретические основы проектов в рамках жизненного цикла;
- методологию проектной деятельности;
- современные концепции, теории, законы и методы в области информатики;
- основные понятия и сопряженные с ними из предметной области;
- систему диагностики и оценки реализованного проекта;

Задания, необходимые для оценивания сформированности УК-2 на пороговом и продвинутом уровне

Написать игру «Парные картинки». Игровое поле разделено на клетки, за каждой из которых скрыта картинка. Картинки - парные, т.е. в игровом поле есть две клетки, в которых находятся одинаковые картинки. В начале игры все клетки «закрыты». Щелчок левой кнопкой мыши «открывает» клетку, в клетке появляется картинка. Теперь надо найти клетку, в которой находится такая же картинка, как и в

открытой клетке. Щелчок по другой клетке открывает вторую картинку. Если картинки в «открытых» клетках одинаковые, то эти клетки «исчезают». Если разные – то клетки остаются открытыми. Очередной щелчок закрывает открытые клетки и открывает следующую. Необходимо отметить, что две открытые клетки закрываются даже в том случае, если открытая картинка такая же, как и одна из двух открытых. Игра заканчивается, когда игрок откроет все парные картинки. Поле 4 на 4.

Ключи к заданию: фрагмент программы:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        LoadGame();
    }
    int[] arr;
    int score = 0;
    int moves = 0;
    int[] RandomIndexes()
    {
        Random r = new Random();
        int max = 37;
        int[] x = new int[16];
        for (int i = 0; i < 8; i++)
        {
            bool contains;
            int next;
            do
            {
                next = r.Next(1,max);
                contains = false;
                for (int index = 0; index < i; index++)
                {
                    int n = x[index];
                    if (n == next)
                    {
                        contains = true;
                        break;
                    }
                }
            } while (contains);
            x[i] = next;
        }
    }
}
```

```

    }
    for (int i= 8;i<16;i++)
    {
        x[i] = x[i-8];
    }
    for (int i = 15; i >= 1; i--)
    {
        int j = r.Next(0,i);

        int tmp = x[j];
        x[j] = x[i];
        x[i] = tmp;
    }
    return x;
}
void LoadGame()
{
    arr =RandomIndexes();
    for(int i=1;i<=gameTable.ColumnCount*gameTable.RowCount;i++)
    {
        Text += (" "+arr[i-1]);
        PictureBox pb = new PictureBox() { Name = "pic" + i,Tag= "Fruits/fru" + arr[i -
1] + ".png" };
        pb.BackColor = Color.Purple;
        pb.Image = Image.FromFile("Fruits/fru0.png");
        pb.SizeMode = PictureBoxSizeMode.CenterImage;
        pb.Dock = DockStyle.Fill;
        pb.Click += new EventHandler(PictureClick);
        gameTable.Controls.Add(pb);
    }
}
bool selected=false;
bool selectWrong = false;
string tag="";
string firstSelected = "";
string secondSelected = "";
void PictureClick(object sender, EventArgs e)
{
    if(!selected)
    {
        if(selectWrong)
        {

```

```

        (gameTable.Controls[firstSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
        (gameTable.Controls[secondSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
        gameTable.Controls[firstSelected].Enabled=true;
        gameTable.Controls[secondSelected].Enabled = true;
        firstSelected = "";
        secondSelected = "";
        selectWrong = false;

    }
    tag = (sender as PictureBox).Tag.ToString();
    (sender as PictureBox).Image = Image.FromFile(tag);
    selected = true;
    firstSelected = (sender as PictureBox).Name;
    (sender as PictureBox).Enabled = false;
}
else
{
    (sender as PictureBox).Enabled = false;
    (sender as PictureBox).Image = Image.FromFile((sender as
PictureBox).Tag.ToString());

    if (tag== (sender as PictureBox).Tag.ToString())
    {
        score++;
        scoreText.Text = "Score = " + score;
        if (score==8)
        {
            MessageBox.Show("Вы отгадали все пары!\nВремя - " + time + "\nХоды
- " + moves, "Конец игры");
        }
    }
    else
    {
        secondSelected = (sender as PictureBox).Name;
        selectWrong = true;
    }
    selected = false;
    Text = score.ToString();
    moves++;
    movesText.Text = "Moves = " + moves;

```

```

    }
}
int time = 0;
private void gameTimer_Tick(object sender, EventArgs e)
{
    time++;
    timeText.Text = "Time = " + time + " secs";
}
}

```

2. Создать класс «Двигатель автомобиля», включающий переменные члены: марка а/м, объем топливного бака, расход бензина на 100 км. Функции-члены:

- Создание и инициализация (конструктор),
- Заправка а/м (количество бензина – в аргументе),
- Расход бензина (пройденный путь – в аргументе),
- Выдача сообщения о том, сколько километров можно проехать на оставшемся бензине,
- Деструктор.

Ключ к заданию: фрагмент программы:

```

class Engine
{
char car[50];
int tank_volume;
int gasoline_consumption;
public:
Engine (const char * car0, int tank_volume0, int gasoline_consumption0)
{
strcpy(car, car0);
tank_volume=tank_volume0;
gasoline_consumption= gasoline_consumption0;
}
char *get_car() {return car; }

int refueling (int refueling_car){return refueling_car;}
int path (int path_car){return path_car;}
void message (int refueling_car, int path_car)
{ int res = (refueling(refueling_car)/(gasoline_consumption)*100);
cout<<"На оставшемся бензине можно проехать "<<res<<endl;
}
~ Engine() {cout<<"0"<<endl;}
};

```

```

int main()
{
Engine car1("Камаз", 60, 10);
Engine car2("Лада Приора", 45, 20);
Engine car3("Лада Гранта", 40, 10);
cout<< car1.get_car()<<endl;
car1.message(25, 32);
cout<< car2.get_car()<<endl;
car2.message(20,20);
cout<< car3.get_car()<<endl;
car3.message(40,7);
return 0;
}

```

Уметь:

- планировать и осуществлять все этапы проекта в рамках жизненного цикла;
- использовать систему диагностики и оценки реализованного проекта;
- профессионально решать задачи, связанные с предметной областью, с учетом современных достижений науки;

Задания, необходимые для оценивания сформированности УК-2 на пороговом и продвинутом уровне

Написать игру «Парные картинки». Игровое поле разделено на клетки, за каждой из которых скрыта картинка. Картинки - парные, т.е. в игровом поле есть две клетки, в которых находятся одинаковые картинки. В начале игры все клетки «закрыты». Щелчок левой кнопкой мыши «открывает» клетку, в клетке появляется картинка. Теперь надо найти клетку, в которой находится такая же картинка, как и в открытой клетке. Щелчок по другой клетке открывает вторую картинку. Если картинки в «открытых» клетках одинаковые, то эти клетки «исчезают». Если разные – то клетки остаются открытыми. Очередной щелчок закрывает открытые клетки и открывает следующую. Необходимо отметить, что две открытые клетки закрываются даже в том случае, если открытая картинка такая же, как и одна из двух открытых. Игра заканчивается, когда игрок откроет все парные картинки. Поле 4 на 4.

Ключи к заданию: фрагмент программы:

```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        LoadGame();
    }
}

```

```

}
int[] arr;
int score = 0;
int moves = 0;
int[] RandomIndexes()
{
    Random r = new Random();
    int max = 37;
    int[] x = new int[16];
    for (int i = 0; i < 8; i++)
    {
        bool contains;
        int next;
        do
        {
            next = r.Next(1,max);
            contains = false;
            for (int index = 0; index < i; index++)
            {
                int n = x[index];
                if (n == next)
                {
                    contains = true;
                    break;
                }
            }
        } while (contains);
        x[i] = next;
    }
    for (int i= 8;i<16;i++)
    {
        x[i] = x[i-8];
    }
    for (int i = 15; i >= 1; i--)
    {
        int j = r.Next(0,i);

        int tmp = x[j];
        x[j] = x[i];
        x[i] = tmp;
    }
    return x;
}

```

```

    }
    void LoadGame()
    {
        arr =RandomIndexes();
        for(int i=1;i<=gameTable.ColumnCount*gameTable.RowCount;i++)
        {
            Text += (" "+arr[i-1]);
            PictureBox pb = new PictureBox() { Name = "pic" + i,Tag= "Fruits/fru" + arr[i -
1] + ".png" };
            pb.BackColor = Color.Purple;
            pb.Image = Image.FromFile("Fruits/fru0.png");
            pb.SizeMode = PictureBoxSizeMode.CenterImage;
            pb.Dock = DockStyle.Fill;
            pb.Click += new EventHandler(PictureClick);
            gameTable.Controls.Add(pb);
        }
    }
    bool selected=false;
    bool selectWrong = false;
    string tag="";
    string firstSelected = "";
    string secondSelected = "";
    void PictureClick(object sender, EventArgs e)
    {
        if(!selected)
        {
            if(selectWrong)
            {
                (gameTable.Controls[firstSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
                (gameTable.Controls[secondSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
                gameTable.Controls[firstSelected].Enabled=true;
                gameTable.Controls[secondSelected].Enabled = true;
                firstSelected = "";
                secondSelected = "";
                selectWrong = false;
            }
            tag = (sender as PictureBox).Tag.ToString();
            (sender as PictureBox).Image = Image.FromFile(tag);
            selected = true;

```

```

        firstSelected = (sender as PictureBox).Name;
        (sender as PictureBox).Enabled = false;
    }
    else
    {
        (sender as PictureBox).Enabled = false;
        (sender as PictureBox).Image = Image.FromFile((sender as
PictureBox).Tag.ToString());

        if (tag == (sender as PictureBox).Tag.ToString())
        {
            score++;
            scoreText.Text = "Score = " + score;
            if (score == 8)
            {
                MessageBox.Show("Вы отгадали все пары!\nВремя - " + time + "\nХоды
- " + moves, "Конец игры");
            }
        }
        else
        {
            secondSelected = (sender as PictureBox).Name;
            selectWrong = true;
        }
        selected = false;
        Text = score.ToString();
        moves++;
        movesText.Text = "Moves = " + moves;
    }
}
int time = 0;
private void gameTimer_Tick(object sender, EventArgs e)
{
    time++;
    timeText.Text = "Time = " + time + " secs";
}
}

```

2. Создать класс «Двигатель автомобиля», включающий переменные члены: марка а/м, объем топливного бака, расход бензина на 100 км. Функции-члены:

- Создание и инициализация (конструктор),
- Заправка а/м (количество бензина – в аргументе),

- Расход бензина (пройденный путь – в аргументе),
- Выдача сообщения о том, сколько километров можно проехать на оставшемся бензине,
- Деструктор.

Ключ к заданию: фрагмент программы:

```
class Engine
{
char car[50];
int tank_volume;
int gasoline_consumption;
public:
Engine (const char * car0, int tank_volume0, int gasoline_consumption0)
{
strcpy(car, car0);
tank_volume=tank_volume0;
gasoline_consumption= gasoline_consumption0;
}
char *get_car() {return car; }

int refueling (int refueling_car){return refueling_car;}
int path (int path_car){return path_car;}
void message (int refueling_car, int path_car)
{ int res = (refueling(refueling_car)/(gasoline_consumption)*100);
cout<<"На оставшемся бензине можно проехать "<<res<<endl;
}
~ Engine() {cout<<"0"<<endl;}
};

int main()
{
Engine car1("Камаз", 60, 10);
Engine car2("Лада Приора", 45, 20);
Engine car3("Лада Гранта", 40, 10);
cout<< car1.get_car()<<endl;
car1.message(25, 32);
cout<< car2.get_car()<<endl;
car2.message(20,20);
cout<< car3.get_car()<<endl;
car3.message(40,7);
return 0;
}
```

Владеть:

- опытом (навыками) использования системы диагностики и оценки реализованного проекта;
- опытом планирования реализации проекта в соответствующей предметной области.

Задания, необходимые для оценивания сформированности УК-2 на продвинутом уровне

Написать игру «Парные картинки». Игровое поле разделено на клетки, за каждой из которых скрыта картинка. Картинки - парные, т.е. в игровом поле есть две клетки, в которых находятся одинаковые картинки. В начале игры все клетки «закрываются». Щелчок левой кнопкой мыши «открывает» клетку, в клетке появляется картинка. Теперь надо найти клетку, в которой находится такая же картинка, как и в открытой клетке. Щелчок по другой клетке открывает вторую картинку. Если картинки в «открытых» клетках одинаковые, то эти клетки «исчезают». Если разные – то клетки остаются открытыми. Очередной щелчок закрывает открытые клетки и открывает следующую. Необходимо отметить, что две открытые клетки закрываются даже в том случае, если открытая картинка такая же, как и одна из двух открытых. Игра заканчивается, когда игрок откроет все парные картинки. Поле 4 на 4.

Ключи к заданию: фрагмент программы:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        LoadGame();
    }
    int[] arr;
    int score = 0;
    int moves = 0;
    int[] RandomIndexes()
    {
        Random r = new Random();
        int max = 37;
        int[] x = new int[16];
        for (int i = 0; i < 8; i++)
        {
            bool contains;
            int next;
            do
            {
```

```

        next = r.Next(1,max);
        contains = false;
        for (int index = 0; index < i; index++)
        {
            int n = x[index];
            if (n == next)
            {
                contains = true;
                break;
            }
        }
        while (contains);
        x[i] = next;
    }
    for (int i= 8;i<16;i++)
    {
        x[i] = x[i-8];
    }
    for (int i = 15; i >= 1; i--)
    {
        int j = r.Next(0,i);

        int tmp = x[j];
        x[j] = x[i];
        x[i] = tmp;
    }
    return x;
}
void LoadGame()
{
    arr =RandomIndexes();
    for(int i=1;i<=gameTable.ColumnCount*gameTable.RowCount;i++)
    {
        Text += (" "+arr[i-1]);
        PictureBox pb = new PictureBox() { Name = "pic" + i,Tag= "Fruits/fru" + arr[i -
1] + ".png" };
        pb.BackColor = Color.Purple;
        pb.Image = Image.FromFile("Fruits/fru0.png");
        pb.SizeMode = PictureBoxSizeMode.CenterImage;
        pb.Dock = DockStyle.Fill;
        pb.Click += new EventHandler(PictureClick);
        gameTable.Controls.Add(pb);
    }
}

```

```

    }
}
bool selected=false;
bool selectWrong = false;
string tag="";
string firstSelected = "";
string secondSelected = "";
void PictureClick(object sender, EventArgs e)
{
    if(!selected)
    {
        if(selectWrong)
        {
            (gameTable.Controls[firstSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
            (gameTable.Controls[secondSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
            gameTable.Controls[firstSelected].Enabled=true;
            gameTable.Controls[secondSelected].Enabled = true;
            firstSelected = "";
            secondSelected = "";
            selectWrong = false;

        }
        tag = (sender as PictureBox).Tag.ToString();
        (sender as PictureBox).Image = Image.FromFile(tag);
        selected = true;
        firstSelected = (sender as PictureBox).Name;
        (sender as PictureBox).Enabled = false;
    }
    else
    {
        (sender as PictureBox).Enabled = false;
        (sender as PictureBox).Image = Image.FromFile((sender as
PictureBox).Tag.ToString());

        if (tag== (sender as PictureBox).Tag.ToString())
        {
            score++;
            scoreText.Text = "Score = " + score;
            if (score==8)
            {

```

```

        MessageBox.Show("Вы отгадали все пары!\nВремя - " + time + "\nХоды - " + moves, "Конец игры");
    }
}
else
{
    secondSelected = (sender as PictureBox).Name;
    selectWrong = true;
}
selected = false;
Text = score.ToString();
moves++;
movesText.Text = "Moves = " + moves;
}
}
int time = 0;
private void gameTimer_Tick(object sender, EventArgs e)
{
    time++;
    timeText.Text = "Time = " + time + " secs";
}
}

```

2. Создать класс «Двигатель автомобиля», включающий переменные члены: марка а/м, объем топливного бака, расход бензина на 100 км. Функции-члены:

- Создание и инициализация (конструктор),
- Заправка а/м (количество бензина – в аргументе),
- Расход бензина (пройденный путь – в аргументе),
- Выдача сообщения о том, сколько километров можно проехать на оставшемся бензине,
- Деструктор.

Ключ к заданию: фрагмент программы:

```

class Engine
{
    char car[50];
    int tank_volume;
    int gasoline_consumption;
public:
    Engine (const char * car0, int tank_volume0, int gasoline_consumption0)
    {
        strcpy(car, car0);
    }
}

```

```

tank_volume=tank_volume0;
gasoline_consumption= gasoline_consumption0;
}
char *get_car() {return car; }

int refueling (int refueling_car){return refueling_car;}
int path (int path_car){return path_car;}
void message (int refueling_car, int path_car)
{ int res = (refueling(refueling_car)/(gasoline_consumption)*100);
cout<<"На оставшемся бензине можно проехать "<<res<<endl;
}
~ Engine() {cout<<"0"<<endl;}
};

int main()
{
Engine car1("Камаз", 60, 10);
Engine car2("Лада Приора", 45, 20);
Engine car3("Лада Гранта", 40, 10);
cout<< car1.get_car()<<endl;
car1.message(25, 32);
cout<< car2.get_car()<<endl;
car2.message(20,20);
cout<< car3.get_car()<<endl;
car3.message(40,7);
return 0;
}

```

Промежуточная аттестация

УК-2 «Способен управлять проектом на всех этапах его жизненного цикла»

Знать:

- теоретические основы проектов в рамках жизненного цикла;
- методологию проектной деятельности;
- современные концепции, теории, законы и методы в области информатики;
- основные понятия и сопряженные с ними из предметной области;
- систему диагностики и оценки реализованного проекта;

Уметь:

- планировать и осуществлять все этапы проекта в рамках жизненного цикла;
- использовать систему диагностики и оценки реализованного проекта;
- профессионально решать задачи, связанные с предметной областью, с учетом современных достижений науки;

Владеть:

- опытом (навыками) использования системы диагностики и оценки реализованного проекта;
- опытом планирования реализации проекта в соответствующей предметной области.

Задания, необходимые для оценивания сформированности УК-2

Перечень задач для зачета

1. Написать программу «Кроссворд».
2. Создать класс «Мобильный телефон», включающий переменные члены: номер, имя владельца, количество денег на счете. Функции-члены:
 - Создание и инициализация (конструктор),
 - Пополнение счета (сумма – в аргументе),
 - Оплата разговоров (тариф и время – в аргументе),
 - Выдача сообщения об остатке средств на счете,
 - Деструктор.

Текущая аттестация

СПК-2 «Способен к преподаванию учебных курсов, дисциплин (модулей) по образовательным программам в образовательных организациях соответствующего уровня образования»

Знать:

- теоретические основы программирования;
- методологию, теорию и эффективную практику образовательной деятельности;
- современные концепции, теории, законы и методы в области информатики и перспективные направления развития современной науки;
- особенности планирования и осуществления учебного процесса в соответствующей предметной области;
- основные понятия и сопряженные с ними из предметной области;
- систему диагностики и оценки уровня образовательных достижений обучающихся;

Задания, необходимые для оценивания сформированности СПК-2 на пороговом и продвинутом уровне

Написать игру «Парные картинки». Игровое поле разделено на клетки, за каждой из которых скрыта картинка. Картинки - парные, т.е. в игровом поле есть две клетки, в которых находятся одинаковые картинки. В начале игры все клетки «закрыты». Щелчок левой кнопкой мыши «открывает» клетку, в клетке появляется картинка. Теперь надо найти клетку, в которой находится такая же картинка, как и в открытой клетке. Щелчок по другой клетке открывает вторую картинку. Если

картинки в «открытых» клетках одинаковые, то эти клетки «исчезают». Если разные – то клетки остаются открытыми. Очередной щелчок закрывает открытые клетки и открывает следующую. Необходимо отметить, что две открытые клетки закрываются даже в том случае, если открытая картинка такая же, как и одна из двух открытых. Игра заканчивается, когда игрок откроет все парные картинки. Поле 4 на 4.

Ключи к заданию: фрагмент программы:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        LoadGame();
    }
    int[] arr;
    int score = 0;
    int moves = 0;
    int[] RandomIndexes()
    {
        Random r = new Random();
        int max = 37;
        int[] x = new int[16];
        for (int i = 0; i < 8; i++)
        {
            bool contains;
            int next;
            do
            {
                next = r.Next(1,max);
                contains = false;
                for (int index = 0; index < i; index++)
                {
                    int n = x[index];
                    if (n == next)
                    {
                        contains = true;
                        break;
                    }
                }
            } while (contains);
            x[i] = next;
        }
    }
}
```

```

    for (int i= 8;i<16;i++)
    {
        x[i] = x[i-8];
    }
    for (int i = 15; i >= 1; i--)
    {
        int j = r.Next(0,i);

        int tmp = x[j];
        x[j] = x[i];
        x[i] = tmp;
    }
    return x;
}
void LoadGame()
{
    arr =RandomIndexes();
    for(int i=1;i<=gameTable.ColumnCount*gameTable.RowCount;i++)
    {
        Text += (" "+arr[i-1]);
        PictureBox pb = new PictureBox() { Name = "pic" + i,Tag= "Fruits/fru" + arr[i -
1] + ".png" };
        pb.BackColor = Color.Purple;
        pb.Image = Image.FromFile("Fruits/fru0.png");
        pb.SizeMode = PictureBoxSizeMode.CenterImage;
        pb.Dock = DockStyle.Fill;
        pb.Click += new EventHandler(PictureClick);
        gameTable.Controls.Add(pb);
    }
}
bool selected=false;
bool selectWrong = false;
string tag="";
string firstSelected = "";
string secondSelected = "";
void PictureClick(object sender, EventArgs e)
{
    if(!selected)
    {
        if(selectWrong)
        {

```

```

        (gameTable.Controls[firstSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
        (gameTable.Controls[secondSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
        gameTable.Controls[firstSelected].Enabled=true;
        gameTable.Controls[secondSelected].Enabled = true;
        firstSelected = "";
        secondSelected = "";
        selectWrong = false;

    }
    tag = (sender as PictureBox).Tag.ToString();
    (sender as PictureBox).Image = Image.FromFile(tag);
    selected = true;
    firstSelected = (sender as PictureBox).Name;
    (sender as PictureBox).Enabled = false;
}
else
{
    (sender as PictureBox).Enabled = false;
    (sender as PictureBox).Image = Image.FromFile((sender as
PictureBox).Tag.ToString());

    if (tag== (sender as PictureBox).Tag.ToString())
    {
        score++;
        scoreText.Text = "Score = " + score;
        if (score==8)
        {
            MessageBox.Show("Вы отгадали все пары!\nВремя - " + time + "\nХоды
- " + moves, "Конец игры");
        }
    }
    else
    {
        secondSelected = (sender as PictureBox).Name;
        selectWrong = true;
    }
    selected = false;
    Text = score.ToString();
    moves++;
    movesText.Text = "Moves = " + moves;

```

```

    }
}
int time = 0;
private void gameTimer_Tick(object sender, EventArgs e)
{
    time++;
    timeText.Text = "Time = " + time + " secs";
}
}

```

2. Создать класс «Двигатель автомобиля», включающий переменные члены: марка а/м, объем топливного бака, расход бензина на 100 км. Функции-члены:

- Создание и инициализация (конструктор),
- Заправка а/м (количество бензина – в аргументе),
- Расход бензина (пройденный путь – в аргументе),
- Выдача сообщения о том, сколько километров можно проехать на оставшемся бензине,
- Деструктор.

Ключ к заданию: фрагмент программы:

```

class Engine
{
char car[50];
int tank_volume;
int gasoline_consumption;
public:
Engine (const char * car0, int tank_volume0, int gasoline_consumption0)
{
strcpy(car, car0);
tank_volume=tank_volume0;
gasoline_consumption= gasoline_consumption0;
}
char *get_car() {return car; }

int refueling (int refueling_car){return refueling_car;}
int path (int path_car){return path_car;}
void message (int refueling_car, int path_car)
{ int res = (refueling(refueling_car)/(gasoline_consumption)*100);
cout<<"На оставшемся бензине можно проехать "<<res<<endl;
}
~ Engine() {cout<<"0"<<endl;}
};

```

```

int main()
{
Engine car1("Камаз", 60, 10);
Engine car2("Лада Приора", 45, 20);
Engine car3("Лада Гранта", 40, 10);
cout<< car1.get_car()<<endl;
car1.message(25, 32);
cout<< car2.get_car()<<endl;
car2.message(20,20);
cout<< car3.get_car()<<endl;
car3.message(40,7);
return 0;
}

```

Уметь:

- планировать и осуществлять все составляющие учебного процесса в соответствии в соответствующей предметной области;
- использовать систему диагностики и оценки уровня образовательных достижений обучающихся;
- профессионально решать задачи, связанные с предметной областью, с учетом современных достижений науки;

Задания, необходимые для оценивания сформированности СПК-2 на пороговом и продвинутом уровне

Написать игру «Парные картинки». Игровое поле разделено на клетки, за каждой из которых скрыта картинка. Картинки - парные, т.е. в игровом поле есть две клетки, в которых находятся одинаковые картинки. В начале игры все клетки «закрыты». Щелчок левой кнопкой мыши «открывает» клетку, в клетке появляется картинка. Теперь надо найти клетку, в которой находится такая же картинка, как и в открытой клетке. Щелчок по другой клетке открывает вторую картинку. Если картинки в «открытых» клетках одинаковые, то эти клетки «исчезают». Если разные – то клетки остаются открытыми. Очередной щелчок закрывает открытые клетки и открывает следующую. Необходимо отметить, что две открытые клетки закрываются даже в том случае, если открытая картинка такая же, как и одна из двух открытых. Игра заканчивается, когда игрок откроет все парные картинки. Поле 4 на 4.

Ключи к заданию: фрагмент программы:

```

public partial class Form1 : Form
{
    public Form1()
    {

```

```

    InitializeComponent();
    LoadGame();
}
int[] arr;
int score = 0;
int moves = 0;
int[] RandomIndexes()
{
    Random r = new Random();
    int max = 37;
    int[] x = new int[16];
    for (int i = 0; i < 8; i++)
    {
        bool contains;
        int next;
        do
        {
            next = r.Next(1,max);
            contains = false;
            for (int index = 0; index < i; index++)
            {
                int n = x[index];
                if (n == next)
                {
                    contains = true;
                    break;
                }
            }
        } while (contains);
        x[i] = next;
    }
    for (int i= 8;i<16;i++)
    {
        x[i] = x[i-8];
    }
    for (int i = 15; i >= 1; i--)
    {
        int j = r.Next(0,i);

        int tmp = x[j];
        x[j] = x[i];
        x[i] = tmp;
    }
}

```

```

    }
    return x;
}
void LoadGame()
{
    arr =RandomIndexes();
    for(int i=1;i<=gameTable.ColumnCount*gameTable.RowCount;i++)
    {
        Text += (" "+arr[i-1]);
        PictureBox pb = new PictureBox() { Name = "pic" + i,Tag= "Fruits/fru" + arr[i -
1] + ".png" };
        pb.BackColor = Color.Purple;
        pb.Image = Image.FromFile("Fruits/fru0.png");
        pb.SizeMode = PictureBoxSizeMode.CenterImage;
        pb.Dock = DockStyle.Fill;
        pb.Click += new EventHandler(PictureClick);
        gameTable.Controls.Add(pb);
    }
}
bool selected=false;
bool selectWrong = false;
string tag="";
string firstSelected = "";
string secondSelected = "";
void PictureClick(object sender, EventArgs e)
{
    if(!selected)
    {
        if(selectWrong)
        {
            (gameTable.Controls[firstSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
            (gameTable.Controls[secondSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
            gameTable.Controls[firstSelected].Enabled=true;
            gameTable.Controls[secondSelected].Enabled = true;
            firstSelected = "";
            secondSelected = "";
            selectWrong = false;
        }
    }
    tag = (sender as PictureBox).Tag.ToString();
}

```

```

        (sender as PictureBox).Image = Image.FromFile(tag);
        selected = true;
        firstSelected = (sender as PictureBox).Name;
        (sender as PictureBox).Enabled = false;
    }
    else
    {
        (sender as PictureBox).Enabled = false;
        (sender as PictureBox).Image = Image.FromFile((sender as
PictureBox).Tag.ToString());

        if (tag == (sender as PictureBox).Tag.ToString())
        {
            score++;
            scoreText.Text = "Score = " + score;
            if (score == 8)
            {
                MessageBox.Show("Вы отгадали все пары!\nВремя - " + time + "\nХоды
- " + moves, "Конец игры");
            }
        }
        else
        {
            secondSelected = (sender as PictureBox).Name;
            selectWrong = true;
        }
        selected = false;
        Text = score.ToString();
        moves++;
        movesText.Text = "Moves = " + moves;
    }
}
}
int time = 0;
private void gameTimer_Tick(object sender, EventArgs e)
{
    time++;
    timeText.Text = "Time = " + time + " secs";
}
}

```

2. Создать класс «Двигатель автомобиля», включающий переменные члены: марка а/м, объем топливного бака, расход бензина на 100 км. Функции-члены:

- Создание и инициализация (конструктор),
- Заправка а/м (количество бензина – в аргументе),
- Расход бензина (пройденный путь – в аргументе),
- Выдача сообщения о том, сколько километров можно проехать на оставшемся бензине,
- Деструктор.

Ключ к заданию: фрагмент программы:

```
class Engine
{
char car[50];
int tank_volume;
int gasoline_consumption;
public:
Engine (const char * car0, int tank_volume0, int gasoline_consumption0)
{
strcpy(car, car0);
tank_volume=tank_volume0;
gasoline_consumption= gasoline_consumption0;
}
char *get_car() {return car; }

int refueling (int refueling_car){return refueling_car;}
int path (int path_car){return path_car;}
void message (int refueling_car, int path_car)
{ int res = (refueling(refueling_car)/(gasoline_consumption)*100);
cout<<"На оставшемся бензине можно проехать "<<res<<endl;
}
~ Engine() {cout<<"0"<<endl;}
};

int main()
{
Engine car1("Камаз", 60, 10);
Engine car2("Лада Приора", 45, 20);
Engine car3("Лада Гранта", 40, 10);
cout<< car1.get_car()<<endl;
car1.message(25, 32);
cout<< car2.get_car()<<endl;
car2.message(20,20);
cout<< car3.get_car()<<endl;
car3.message(40,7);
}
```

```
return 0;  
}
```

Владеть:

- опытом (навыками) использования системы диагностики и оценки уровня образовательных достижений обучающихся;
- опытом планирования учебного процесса в соответствующей предметной области;

Задания, необходимые для оценивания сформированности СПК-2 на продвинутом уровне

Написать игру «Парные картинки». Игровое поле разделено на клетки, за каждой из которых скрыта картинка. Картинки - парные, т.е. в игровом поле есть две клетки, в которых находятся одинаковые картинки. В начале игры все клетки «закрываются». Щелчок левой кнопкой мыши «открывает» клетку, в клетке появляется картинка. Теперь надо найти клетку, в которой находится такая же картинка, как и в открытой клетке. Щелчок по другой клетке открывает вторую картинку. Если картинки в «открытых» клетках одинаковые, то эти клетки «исчезают». Если разные – то клетки остаются открытыми. Очередной щелчок закрывает открытые клетки и открывает следующую. Необходимо отметить, что две открытые клетки закрываются даже в том случае, если открытая картинка такая же, как и одна из двух открытых. Игра заканчивается, когда игрок откроет все парные картинки. Поле 4 на 4.

Ключи к заданию: фрагмент программы:

```
public partial class Form1 : Form  
{  
    public Form1()  
    {  
        InitializeComponent();  
        LoadGame();  
    }  
    int[] arr;  
    int score = 0;  
    int moves = 0;  
    int[] RandomIndexes()  
    {  
        Random r = new Random();  
        int max = 37;  
        int[] x = new int[16];  
        for (int i = 0; i < 8; i++)  
        {
```

```

    bool contains;
    int next;
    do
    {
        next = r.Next(1,max);
        contains = false;
        for (int index = 0; index < i; index++)
        {
            int n = x[index];
            if (n == next)
            {
                contains = true;
                break;
            }
        }
    } while (contains);
    x[i] = next;
}
for (int i = 8;i<16;i++)
{
    x[i] = x[i-8];
}
for (int i = 15; i >= 1; i--)
{
    int j = r.Next(0,i);

    int tmp = x[j];
    x[j] = x[i];
    x[i] = tmp;
}
return x;
}
void LoadGame()
{
    arr =RandomIndexes();
    for(int i=1;i<=gameTable.ColumnCount*gameTable.RowCount;i++)
    {
        Text += (" "+arr[i-1]);
        PictureBox pb = new PictureBox() { Name = "pic" + i,Tag= "Fruits/fru" + arr[i -
1] + ".png" };
        pb.BackColor = Color.Purple;
        pb.Image = Image.FromFile("Fruits/fru0.png");
    }
}

```

```

        pb.SizeMode = PictureBoxSizeMode.CenterImage;
        pb.Dock = DockStyle.Fill;
        pb.Click += new EventHandler(PictureClick);
        gameTable.Controls.Add(pb);
    }
}
bool selected=false;
bool selectWrong = false;
string tag="";
string firstSelected = "";
string secondSelected = "";
void PictureClick(object sender, EventArgs e)
{
    if(!selected)
    {
        if(selectWrong)
        {
            (gameTable.Controls[firstSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
            (gameTable.Controls[secondSelected] as PictureBox).Image=
Image.FromFile("Fruits/fru0.png");
            gameTable.Controls[firstSelected].Enabled=true;
            gameTable.Controls[secondSelected].Enabled = true;
            firstSelected = "";
            secondSelected = "";
            selectWrong = false;

        }
        tag = (sender as PictureBox).Tag.ToString();
        (sender as PictureBox).Image = Image.FromFile(tag);
        selected = true;
        firstSelected = (sender as PictureBox).Name;
        (sender as PictureBox).Enabled = false;
    }
    else
    {
        (sender as PictureBox).Enabled = false;
        (sender as PictureBox).Image = Image.FromFile((sender as
PictureBox).Tag.ToString());

        if (tag==(sender as PictureBox).Tag.ToString())
        {

```

```

        score++;
        scoreText.Text = "Score = " + score;
        if (score==8)
        {
            MessageBox.Show("Вы отгадали все пары!\nВремя - " + time + "\nХоды
- " + moves, "Конец игры");
        }
    }
    else
    {
        secondSelected = (sender as PictureBox).Name;
        selectWrong = true;
    }
    selected = false;
    Text = score.ToString();
    moves++;
    movesText.Text = "Moves = " + moves;
}
}
int time = 0;
private void gameTimer_Tick(object sender, EventArgs e)
{
    time++;
    timeText.Text = "Time = " + time + " secs";
}
}

```

2. Создать класс «Двигатель автомобиля», включающий переменные члены: марка а/м, объем топливного бака, расход бензина на 100 км. Функции-члены:

- Создание и инициализация (конструктор),
- Заправка а/м (количество бензина – в аргументе),
- Расход бензина (пройденный путь – в аргументе),
- Выдача сообщения о том, сколько километров можно проехать на оставшемся бензине,
- Деструктор.

Ключ к заданию: фрагмент программы:

```

class Engine
{
    char car[50];
    int tank_volume;
    int gasoline_consumption;
}

```

```

public:
Engine (const char * car0, int tank_volume0, int gasoline_consumption0)
{
strcpy(car, car0);
tank_volume=tank_volume0;
gasoline_consumption= gasoline_consumption0;
}
char *get_car() {return car; }

int refueling (int refueling_car){return refueling_car;}
int path (int path_car){return path_car;}
void message (int refueling_car, int path_car)
{ int res = (refueling(refueling_car)/(gasoline_consumption)*100);
cout<<"На оставшемся бензине можно проехать "<<res<<endl;
}
~ Engine() {cout<<"0"<<endl;}
};

int main()
{
Engine car1("Камаз", 60, 10);
Engine car2("Лада Приора", 45, 20);
Engine car3("Лада Гранта", 40, 10);
cout<< car1.get_car()<<endl;
car1.message(25, 32);
cout<< car2.get_car()<<endl;
car2.message(20,20);
cout<< car3.get_car()<<endl;
car3.message(40,7);
return 0;
}

```

Промежуточная аттестация

СПК-2 «Способен к преподаванию учебных курсов, дисциплин (модулей) по образовательным программам в образовательных организациях соответствующего уровня образования»

Знать:

- теоретические основы программирования;
- методологию, теорию и эффективную практику образовательной деятельности;

- современные концепции, теории, законы и методы в области информатики и перспективные направления развития современной науки;
- особенности планирования и осуществления учебного процесса в соответствующей предметной области;
- основные понятия и сопряженные с ними из предметной области;
- систему диагностики и оценки уровня образовательных достижений обучающихся;

Уметь:

- планировать и осуществлять все составляющие учебного процесса в соответствии в соответствующей предметной области;
- использовать систему диагностики и оценки уровня образовательных достижений обучающихся;
- профессионально решать задачи, связанные с предметной областью, с учетом современных достижений науки;

Владеть:

- опытом (навыками) использования системы диагностики и оценки уровня образовательных достижений обучающихся;
- опытом планирования учебного процесса в соответствующей предметной области.

Задания, необходимые для оценивания сформированности СПК-2

Перечень задач для зачета

3. Написать программу «Кроссворд».
4. Создать класс «Мобильный телефон», включающий переменные члены: номер, имя владельца, количество денег на счете. Функции-члены:
 - Создание и инициализация (конструктор),
 - Пополнение счета (сумма – в аргументе),
 - Оплата разговоров (тариф и время – в аргументе),
 - Выдача сообщения об остатке средств на счете,
 - Деструктор.

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедура оценивания знаний и умений состоит из следующих составных элементов: учета посещаемости лекционных занятий, выполнения лабораторных работ, выполнения самостоятельных практических работ.

Требования к выполнению лабораторных работ

Лабораторные работы включают следующие части: подготовка к работе; выполнение и оформление работы; сдача отчета работы.

Описание работы излагается в виде четкого алгоритма и выдается каждому обучающемуся.

Подготовка обучающихся к работе состоит в повторении теоретического материала. Кроме того, обучающиеся знакомятся с описанием лабораторной работы, изучают ее цели, содержание, письменно ответить на контрольные вопросы, делают все необходимые записи в соответствии с требованиями к оформлению отчета по работе.

В зависимости от содержания лабораторной работы требования к оформлению отчетов могут быть различными.

Лабораторная работа считается выполненной, если: предоставлен отчет о результатах выполнения задания.

Требования к оформлению и выполнению работы определены в методических рекомендациях.

Требования к выполнению самостоятельных практических работ

Выполнение самостоятельных практических работ (выполнение дополнительных заданий) направлено на углубление и расширение знаний по соответствующей тематике. Самостоятельная практическая работа считается выполненной, если предоставлен отчет в соответствии с требованиями. Требования к оформлению и выполнению работы определены в методических рекомендациях.

Освоение дисциплины оценивается по балльной шкале. Общее количество баллов по дисциплине - 100 баллов.

Максимальное количество баллов, которое можно набрать в течение семестра за посещаемость, выполнение лабораторных работ и самостоятельных работ - 80 баллов.

За посещение лекционных занятий и лабораторных работ магистрант может набрать максимально до 2 баллов.

За выполнение лабораторных работ магистрант может набрать максимально 40 баллов (всего 4 лабораторных работ, по 10 баллов за одну работу).

За выполнение самостоятельных практических работ магистрант может набрать максимально 40 баллов (всего 5 самостоятельных практических работ, по 8 баллов за один конспект).

Обучающийся, набравший 41 балл и более, допускается к зачету.

Максимальная сумма баллов, которые магистрант может набрать при сдаче зачета, составляет 20 баллов.

Требования к зачету

Студент до зачета должен:

- посетить лекции и лабораторные работы;
- отчитаться по темам для самостоятельного изучения.

Шкала оценивания зачета

Критерии оценивания	Баллы
В практическом задании написан алгоритм с некоторыми неточностями и реализован фрагмент программы.	5
В практическом задании полностью написан алгоритм и реализована программа с существенными ошибками.	10
В практическом задании полностью написан алгоритм, полностью реализована программа, которая запускается, но работает не совсем корректно с учетом условия.	15
В практическом задании полностью написан алгоритм и полностью написана программа, которая работает, но в ней могут иметься некоторые несущественные неточности.	20

Итоговая шкала оценивания результатов освоения дисциплины

Итоговая оценка по дисциплине формируется из суммы баллов по результатам текущего контроля и промежуточной аттестации и выставляется в соответствии с приведенной ниже таблицей.

Оценка по 100-балльной системе	Оценка по 5-балльной системе
81 – 100	Отлично
61 – 80	Хорошо
41 – 60	Удовлетворительно
0 – 40	Неудовлетворительно

