

Министерство образования и науки Российской Федерации  
 Федеральное государственное бюджетное образовательное учреждение  
 высшего образования  
 «Владимирский государственный университет  
 имени Александра Григорьевича и Николая Григорьевича Столетовых»  
 (ВлГУ)



УТВЕРЖДАЮ

Проректор по УМР

А.А.Панфилов

«17» 03 2016 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
 СОВРЕМЕННЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Направление подготовки 44.03.05 – Педагогическое образование

Профили подготовки Информатика. Математика

Уровень высшего образования бакалавриат

Форма обучения очная

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточного контроля (экз./зачет)
3	4 / 144	18	18	36	72	зачет (с оценкой)
4	5 / 180	18	18	36	108	зачет
5	5 / 180	18	18	18	81	экзамен (45)
Итого	14/ 504	54	54	90	261	зачет (с оценкой) / зачет / экзамен (45)

Владимир 2016

## 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целями освоения дисциплины «Современные языки программирования» являются:

- изучение основных концепций структурного, объектно-ориентированного и функционального программирования;
- развитие логического и алгоритмического мышления у студентов;
- формирование и развитие умений планирования действий и структурирования информации, необходимых для решения поставленной задачи, при помощи фиксированного набора средств;
- формирование способности отображать реальные объекты и явления в компьютерные информационные структуры;
- использование языка программирования как средства автоматизации вычислений, необходимых в учебной деятельности;
- расширение знаний о языках программирования;
- формирование умений разработки визуальных приложений.

В качестве объектно-ориентированного языка программирования рассматривается язык C#, который в настоящий момент является одним из основных языков разработки программного обеспечения. В школьном курсе информатики данный язык программирования не всегда рассматривается, однако на уровне консольных приложений алгоритмические конструкции языка C# идентичны конструкциям языка C++, который включен в перечень языков, используемых в Едином государственном экзамене по Информатике и ИКТ, а также на Всероссийской олимпиаде школьников по информатике. Таким образом, знание этих языков является необходимым для современного учителя информатики. В качестве языка функционального программирования рассматривается F#.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Современные языки программирования» является курсом по выбору.

В ходе изучения дисциплины студенты применяют знания и умения, сформированные при изучении дисциплины «Программирование», учатся переводить решения задач с одного языка программирования на другой, тем самым прослеживая сходства и различия между языками программирования и отслеживая тенденции в развитии языков программирования.

Для успешного освоения содержания дисциплины студент должен знать:

- один из языков программирования;
- основные управляющие алгоритмические конструкции;

- основы объектно-ориентированного программирования.

Знания, умения и практические навыки, приобретенные студентами в результате изучения дисциплины «Современные языки программирования», будут использоваться при освоении дисциплин: «Компьютерное моделирование», «Частные методики», «Методика обучения информатике», а также при прохождении педагогической практики.

### **3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

Дисциплина «Современные языки программирования» нацелена на формирование у выпускника следующих компетенций:

- способность к самоорганизации и к самообразованию (ОК-6);
- готовность реализовывать образовательные программы по учебным предметам в соответствии с требованиями профессиональных стандартов (ПК-1);
- способность руководить учебно-исследовательской деятельностью обучающихся (ПК-12).

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

Знать:

- основные концепции объектно-ориентированного программирования и функционального программирования (ПК-1);
- структуру программы и основные операторы языков программирования C# (C++) и F# (ПК-1, ПК-12);
- основные типы данных и управляющие конструкции языка C# (C++) (ПК-1);
- основные компоненты, используемые при разработке визуальных приложений (ОК-6, ПК-12).

Уметь:

- создавать собственные объекты и классы объектов в C# (ПК-1, ПК-12);
- разрабатывать внешний интерфейс приложения для решения задачи (ПК-12);
- создавать функции (в том числе рекурсивные функции и функции высших порядков) с использованием языка F# (ПК-12);
- работать с кортежами, списками, деревьями средствами языка F# (ПК-12).

Владеть:

- основными методами проектирования и отладки алгоритмов (ОК-6, ПК-1);

- навыками проектирования программ на языке программирования высокого уровня (ОК-6, ПК-1).

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 14 зачетных единиц, 504 часа.

№	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем УР с применением интерактивных методов	Формы ТКУ, форма ПА
				Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
	<b>Семестр</b>	<b>3</b>									
1.	Основы C++. Ввод-вывод информации. Описание переменных	3	1	2	2	2		4		2 (33%)	
2.	Операторы ветвления и выбора	3	2	2	2	4		8		2 (25%)	
3.	Циклы с предусловием и постусловием	3	3-4	4	4	8		10		8 (50%)	
4.	Массивы. Линейные массивы	3	5-7	2	2	4		10		4 (50%)	Рейтинг-контроль №1
5.	Алгоритмы поиска и сортировок	3	7		2	2		8		2 (50%)	
6.	Массивы. Двумерные массивы	3	8-9	2	2	4		8		2 (25%)	
7.	Строки	3	10-11	2	2	6		8	8	4 (40%)	Рейтинг-контроль №2
8.	Функции пользователя	3	12-13	2	2	4		10		4 (50%)	
9.	Перегрузка функций	3	14	2		2		6		2 (50%)	
	<b>Итого в 3 семестре</b>	<b>3</b>		<b>18</b>	<b>18</b>	<b>36</b>		<b>72</b>		<b>30 (42%)</b>	<b>зачет (с оценкой)</b>
	<b>Семестр</b>	<b>4</b>									
10.	Введение в Visual Studio.NET, Frame.NET. Язык программирования C#.	4	1	2		2		6		2 (50%)	

	Ввод-вывод информации										
11	Понятия класса и объекта. Конструкторы и деструкторы класса	4	2		2	2		6		2 (50%)	
12	Методы класса. Управление доступом к членам класса	4	3	2		2		6		0 (0%)	
13	Перегрузка методов и конструкторов	4	4		2	2		6		2 (50%)	
14	Визуальное программирование. Объекты, свойства, методы и события.	4	5-6	2	2	4		8		4 (50%)	Рейтинг-контроль №1
15	Визуальное программирование. Компоненты Button, Label, TextBox	4	7-8	2	2	4		10		2 (25%)	
16	Визуальное программирование. Компоненты CheckBox, RadioButton	4	9-10	2	2	4		10		2 (25%)	
17	Создание меню приложения. Компоненты MenuStrip, ContextMenuStrip	4	11-12	2	2	4		12		2 (25%)	Рейтинг-контроль №2
18	Визуальное программирование. Компонент Chart	4	13-14	2	2	4		12		4 (50%)	
19	Графические возможности среды. Компонент PictureBox	4	15-16	2	2	4		16		2 (25%)	
20	Анимация в среде программирования C#	4	17-18	2	2	4		16		4 (50%)	Рейтинг-контроль №3
	<b>Итого в 4 семестре</b>			<b>18</b>	<b>18</b>	<b>36</b>		<b>108</b>		<b>38 (53%)</b>	<b>зачет</b>
	<b>Семестр</b>	<b>5</b>									
21	Языки функционального программирования	5	1-2	2	2	2		10		2 (33%)	
22	Язык программирования F#. Анонимные функции. Сопоставление с образцом	5	3-4	2	2	2		10		2 (33%)	
23	Рекурсия	5	5-6	2	2	2		10		2 (33%)	Рейтинг-контроль №1

24	Цикл с аккумулятором	5	7-8	2	2	2		10		2 (33%)	
25	Списки. Операции со списками	5	9-10	2	2	2		10		2 (33%)	
26	Переменные в языке F#. Массивы	5	11-12	2	2	2		10		2 (33%)	Рейтинг-контроль №2
27	Двоичные деревья	5	13-14	2	2	2		10		2 (33%)	
28	Λ-исчисление	5	15-18	4	4	4		20		4 (33%)	Рейтинг-контроль № 3
	<b>Итого в 5 семестре</b>			<b>18</b>	<b>18</b>	<b>18</b>		<b>90</b>		<b>18 (33%)</b>	<b>экзамен (45)</b>
	<b>Итого по курсу</b>			<b>54</b>	<b>54</b>	<b>90</b>		<b>261</b>		<b>86 (43%)</b>	<b>зачет с оценкой / зачет / экзамен (45)</b>

## 5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Процесс изложения учебного материала на лекции сопровождается демонстрацией компьютерной презентации по рассматриваемой теме, а также выполнением интерактивных заданий при наличии требуемого технического обеспечения (интерактивной доски). Для изучения новых разделов проводятся проблемные лекции.

На практических занятиях проводятся групповые консультации, мозговые штурмы, используются разыгрывание ролей (выступление студента в роли школьного учителя). Решение задач выполняется в двух режимах: с использованием компьютера и последующей отладкой программы и проведением эксперимента по работоспособности программ при различных исходных данных, и без использования компьютера с дальнейшей имитацией работы компьютера при наличии созданной программы.

На лабораторных занятиях могут проводиться тренировки на машинных моделях с использованием компьютерных моделей различной степени сложности.

В рамках изучения дисциплины осуществляется поддержка студентов с использованием электронных образовательных технологий (размещение учебных материалов на сайте кафедры), организуются консультации с использованием дистанционных образовательных технологий

## 6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Текущий контроль успеваемости проводится в форме рейтинг-контроля в соответствии с планом работы три раза в семестр. В рейтинг-контроль включаются следующие аспекты работы студента:

- выполнение и защита лабораторных работ;

- участие в устных опросах и коллоквиумах;
- выполнение письменных работ;
- выполнение дополнительных заданий в рамках аудиторной или самостоятельной работы.

По результатам изучения дисциплины в 3 семестре выставляется зачет с оценкой, в 4 семестре - зачет, в 5 семестре – экзамен.

По завершении изучения объектно-ориентированного программирования студенты создают и защищают проекты – самостоятельно разработанные приложения в соответствии с поставленными задачами. По завершении изучения функционального программирования выполняется контрольная работа.

### Примерные задания рейтинг-контролей

#### 3 семестр

#### Рейтинг-контроль №1

1. Записать синтаксис команды ветвления в языке C#
2. Определить значение переменных a, b, c в результате выполнения фрагмента алгоритма

```
int a=8, b=9;
double c;
if (a>b) && (a%2==0)
{
    a-=4;
    b=b+a/2;
    c=a+Math.Sqrt(b-2);
}
else
{
    b+=5;
    a=b-a/2;
    c=a+Math.Pow(b, 2);
}
```

3. Заменить команду ветвления тернарным оператором

```
if (x!=y)
    z=Math.Abs(x-y);
else
    z=0;
```

4. Приведите пример цикла с параметром, который не выполнится ни одного раза.

5. Приведите пример бесконечного цикла с предусловием.

6. Дан фрагмент программы

```
int a, s=0;
for (int i=5; i<=15; i++)
{
    a=int.Parse(Console.ReadLine());
    s+=a;
}
Console.WriteLine(s);
```

Перепишите алгоритм, заменив цикл for

а) циклом с предусловием

б) циклом с постусловием

так, чтобы результат работы алгоритма не изменился.

7. Определите значение переменных s и t в конце работы алгоритма

```
int s, t, n;
s=5; t=1; n=2;
while (n<=4)
{ s=s+(n-1); t=t+s/2; n++;}
```

### Рейтинг-контроль №2

1. Приведите пример цикла с параметром, который не выполнится ни одного раза.
2. Приведите пример бесконечного цикла с предусловием.
3. Дан фрагмент программы

```
int a, s=0;
for (int i=5; i<=15; i++)
{
    a=int.Parse(Console.ReadLine());
    s+=a;
}
Console.WriteLine(s);
```

Перепишите алгоритм, заменив цикл for

а) циклом с предусловием

б) циклом с постусловием

так, чтобы результат работы алгоритма не изменился.

4. Определите значение переменных s и t в конце работы алгоритма

```
int s, t, n;
s=5; t=1; n=2;
while (n<=4)
{ s=s+(n-1); t=t+s/2; n++;}
```

5. Дан алгоритм, который, получив на вход число x, печатает числа a и b. Укажите наименьшее положительное пятизначное число x, при вводе которого алгоритм печатает сначала 6, а потом 3.

```
int x, y, a, b;
a = 0;
b = 10;
x=int.Parse(Console.ReadLine());
while (x > 0)
{
    y = x % 10;    x = x / 10;
    if (y > a)    a = y;
    if (y < b)    b = y;
}
Console.WriteLine(a+" "+b);
```

### Рейтинг-контроль №3

1. Описать массив из 15 целых чисел. Задать им значения 10, 15, 20, 25, ..., 80.
2. В программе описан одномерный целочисленный массив с индексами от 0 до 10 и целочисленные переменные k, i. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for (int i=0; i<=10; i++) A[i]=i;
for (int i=0; i<=4; i++)
{
    k=A[2*i];
    A[2*i]=A[2*i+1];
    A[2*i+1]=k;
}
```

Чему будут равны элементы этого массива?

3. В программе описан одномерный целочисленный массив A с индексами от 0 до 10. Ниже представлен фрагмент этой программы, в котором значения элементов массива сначала задаются, а затем меняются.

```
for (i=0; i<=10; i++)
```



```

    A[i]=i-1;
for (i=1; i<=10; i++)
    A[i-1]=A[i];
A[10]=10;

```

Записать элементы массива после его задания и после преобразования. Определить, как изменятся элементы этого массива после выполнения фрагмента программы (выбрать один из вариантов)?

- 1) все элементы, кроме последнего, окажутся равны между собой
  - 2) все элементы окажутся равны своим индексам
  - 3) все элементы, кроме последнего, будут сдвинуты на один элемент вправо
  - 4) все элементы, кроме последнего, уменьшатся на единицу
4. В таблице Dat представлены данные о количестве голосов, поданных за 10 исполнителей народных песен (Dat[1] – количество голосов, поданных за первого исполнителя, Dat[2] – за второго и т. д.). Определите, какое число будет напечатано в результате работы следующей программы.

```

int m=0, n=0;
int [] Dat=new int[10];
Dat[0]= 35;    Dat[1]= 16;    Dat[2]= 20;
Dat[3]= 20;    Dat[4]= 41;    Dat[5]= 14;
Dat[6]= 21;    Dat[7]= 28;    Dat[8]= 12;
Dat[9]= 25;
for (int k=0; k<10; k++)
if (Dat[k]<25) {m++; n=k;}
Console.WriteLine(n);

```

5. Дан массив чисел: 5, 6, 8, -3, 7, -2, 4. Пошагово отсортировать элементы массива методом простого обмена (пузырьковым).

#### 4 семестр

##### Рейтинг-контроль №1

1. Запишите синтаксис описания метода в C#
2. Запишите особенности передачи параметров метода по значению.
3. Определить, что будет выведено на экране в результате выполнения программы

```

class program
{
    static void p(ref int a, int b)
    {
        a = a * 2;
        b = a % b;
        Console.WriteLine(a + " " + b);
    }
    static void Main(string[] args)
    {
        int x = 10, y = 3;
        p(ref x, y);
        Console.WriteLine(x + " " + y);
    }
}

```

4. Описать класс simple. Полем класса является целое число. Конструктор по умолчанию задает полю значение 2. Конструктор с параметрами задает полю значение, введенное с клавиатуры. Класс имеет 2 метода: 1) проверяет, является ли экземпляр класса простым числом; 2) выводит все простые числа, не превосходящие экземпляр класса .

##### Рейтинг-контроль №2

1. На форме расположены три текстовых поля и кнопка. В первое и третье поля вводятся числа, во второе поле – один из знаков арифметических действий (+, -, \*, /, ^). По нажатию кнопки происходит вычисление значения полученного арифметического выражения.
2. Доработать задачу 1 так, чтобы вычисление арифметического выражения происходило при изменении любого текстового поля. Значения по умолчанию – в числовых полях 0, в поле арифметической операции – «+».
3. На форме расположены два текстовых поля и две кнопки. В поля вводятся числа  $a$  и  $b$ . Первая кнопка осуществляет вывод на форму значений функции  $y = x^2$ , вторая – значений функции  $y = \sqrt{x}$ .
4. На форме расположены два текстовых поля и надпись. В поля вводятся числа  $a$  и  $b$  (в момент начала работы программы в полях находятся соответственно значения 1 и 0). Программа должна находить решения уравнения  $a|x| = b$  и выводить найденное решение в надпись.

### Рейтинг-контроль №3

1. Составить программу, которая рисует ломаную линию по заданным точкам. Координаты вершин ломаной задаются щелчком кнопки мыши по области рисования.
2. Написать программу для построения графика функции  $y = 2x^3 + 3x - 4$  на промежутке  $x \in [-4, 8]$
3. В массиве хранятся  $N$  чисел (вводятся с клавиатуры или генерируются случайным образом). На основе имеющихся в массиве данных построить столбчатую или круговую диаграмму. Выбор типа диаграммы осуществляется с помощью переключателей RadioButton.
4. Реализовать модель звездного неба, предусмотрев в обработку следующих действий:
  - a. появление «звезд» случайным образом
  - b. исчезание звезд случайным образом (для этого можно в случайной области рисунка рисовать небольшой квадрат цветом фона)
  - c. «взрывы» звезд (рисование нескольких концентрических окружностей). Взрывы звезд не должны происходить часто (т.е. не при каждом вызове процедуры обработки таймера).

### 5 семестр

#### Рейтинг-контроль №1

1. Что такое парадигма программирования?
2. Какие парадигмы программирования Вы знаете?
3. В чем заключается функциональная парадигма программирования?
4. Для решения каких задач больше всего подходит функциональное программирование?
5. Перечислите основные достоинства и недостатки функциональных языков программирования.
6. Какие методы конструирования функций Вы знаете?
7. Что такое каррирование?
8. Какие языки функционального программирования Вы знаете?
9. Что нового появилось в языке Haskell?
10. В чем, на Ваш взгляд, состоит причина всплеска интереса к функциональным языкам программирования?
11. Кратко охарактеризуйте язык F#.
12. В чем заключается мультипарадигмальный подход к программированию.

#### Рейтинг-контроль №2

1. В чем особенность синтаксиса F#?
2. Какие типы данных существуют в F#?

3. Что такое опциональные типы данных и зачем они нужны?
4. Что такое литерал в F#?
5. Дайте определение понятию функционального типа.
6. Что такое рекурсия?
7. Что такое прямая и косвенная рекурсия?
8. Каков синтаксис условного оператора и оператора циклов в F#?
9. Напишите функцию, которая по двум числам, представляющим собой коэффициенты уравнения  $ax + b = 0$ , возвращает пару первый элемент которой принимает значение истина, если уравнение имеет решение и ложь, в противном случае, а второй элемент является решением уравнения или равен 0, если решение не существует.
10. Напишите функцию, которая принимает значение истина, если два отрезка, концы которых  $(x_i, y_i)$ , где  $i = 1, 2, 3, 4$  параллельны или лежат на одной прямой.

### Рейтинг-контроль №3

1. Что такое рекурсивные структуры данных?
2. Что представляет собой синтаксис объявления списков?
3. Перечислите способы задания списков.
4. Как происходит сопоставление с образцом для списков?
5. Перечислите функции высших порядков для списков.
6. В чем суть отображения для списков?
7. Определите список и примените к нему функцию отображения.
8. Определите список и отфильтруйте его по заданному критерию.
9. В чем суть операций свертки и редуцирования?
10. Какие виды массивов существуют в F#?
11. В чем особенность структуры данных «дерево»?
12. Перечислите виды деревьев F#.
13. Определите функцию, принимающую на вход целое число  $n$  и возвращающую список, содержащий  $n$  элементов, упорядоченных по возрастанию.
  - a. Список натуральных чисел.
  - b. Список четных натуральных чисел.
  - c. Список квадратов натуральных чисел.
  - d. Список степеней двойки.
  - e. Список пирамидальных чисел. Пирамидальные числа строятся по правилу:  $p_1 = 1$ ,  $p_n = t_n + p_{n-1}$ , где  $t_n - n$ -е треугольное число.

### Примерные задания для самостоятельной работы студентов

#### 3 семестр

1. Составьте программу для решения системы двух линейных уравнений

$$\begin{cases} ax + by = c, \\ dx + ey = f \end{cases}$$

с двумя неизвестными  $x, y$ . Значения неизвестных находятся по формулам

$$\Delta = ae - bd, \quad x = \frac{ce - bf}{\Delta}, \quad y = \frac{af - cd}{\Delta}.$$

2. Написать программу, которая по координатам вершин треугольника находит длины его медиан.
3. Написать программу, которая решает уравнение  $a|x| = b$  относительно  $x$  для любых чисел  $a$  и  $b$ , введенных с клавиатуры. Все числа считаются действительными.
4. Зная разложение функции в ряд по степеням  $x$ , написать программу для нахождения значения этой функции для переменной  $x$  из указанного числового промежутка с точностью до переменной  $e$ .

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \frac{(-1)^n x^{2n}}{(2n)!} + \dots, \quad x - \text{любое, } e = 10^{-8}$$

5. Дан целочисленный массив из  $n$  элементов. Элементы могут принимать значения от 160 до 200 – рост учащихся выпускного класса. В баскетбольную команду берут тех, чей рост не менее 180 см. Напишите программу, которая подсчитывает и выводит минимальный рост игрока баскетбольной команды. Гарантируется, что хотя бы один ученик играет в баскетбольной команде.
6. Напишите программу, которая заменяет каждый элемент матрицы целых чисел размерами  $m \times n$  на сумму элементов его “креста”, т.е. тех элементов, которые находятся в одном с ним столбце и в одной строке.
7. Дан прямоугольный массив целых положительных чисел  $10 \times 20$ . Напишите программу для поиска строки с наименьшей суммой элементов. Вывести на печать номер строки и сумму ее элементов. Предполагается, что такая строка единственна.
8. Даны две строки  $a$  и  $b$ . Напишите программу, которая определяет, можно ли из букв, входящих в строку  $a$ , составить строку  $b$  (буквы можно переставлять, но каждую букву можно использовать не более одного раза.)
9. Дан текст, состоящий из прописных букв русского алфавита, в котором нет условных обозначений и сокращений. Этот текст шифруется по двум ключам:  $N1$  и  $N2$  ( $N1, N2$  – натуральные числа, которые вводятся с клавиатуры), т. е. код первой буквы сдвигается на  $N1$  (символов в алфавите), код второй буквы на  $N2$ , код третьей буквы на  $N1$  символов и т.д. Например, при  $N1 = 2, N2 = 3$  слово «код» преобразуется в слово «мсё». Пробелы и знаки препинания не шифруются. Напишите программу для шифровки и дешифровки предложенного сообщения.

#### 4 семестр

1. Создать главное меню с командами Треугольник и Программа. Меню Треугольник содержит пункты: Ввод, Рисование, Вычисление.  
 Пункт Ввод обеспечивает отображение на экране 6 текстовых полей для ввода координат вершин треугольника.  
 Пункт Рисование обеспечивает отображение картинки, содержащей чертеж построенного по введенным координатам треугольника.  
 Пункт Вычисление обеспечивает появление на форме двух флажков. При отмеченном первом флажке на форму выводится значение периметра треугольника, при отмеченном втором флажке – площадь треугольника.  
 Пункт меню Программа содержит две команды: Выход (которая закрывает форму) и О разработчике (которая выводит информацию об авторе проекта).
2. Создать главное меню с командами Прямоугольник и Программа. Меню Прямоугольник содержит пункты: Размер, Рисование, Оформление.  
 Пункт Размер обеспечивает отображение на экране 2 текстовых полей для ввода ширины и высоты прямоугольника.  
 Пункт Рисование обеспечивает отображение картинки, содержащей прямоугольник с указанными размерами, расположенный в центре графического поля.  
 Пункт Оформление обеспечивает появление на форме двух флажков, содержащей возможность изменять стиль прямоугольника: закрашенный прямоугольник и (или) контур прямоугольника.  
 Пункт меню Программа содержит две команды: Выход (которая закрывает форму) и О разработчике (которая выводит информацию об авторе проекта).
3. Создать главное меню с командами График и Программа. Меню График содержит пункты: Диапазон, Функция, Таблица.  
 Пункт Диапазон обеспечивает отображение на экране 2 текстовых полей для ввода диапазона значений аргумента функции и построение графика функции для указанного диапазона значений.

Пункт Функция обеспечивает отображение двух переключателей, позволяющих выбрать одну из двух доступных для построения графика функций:  $y = 2x^3 - 4x^2 + x - 2$  или  $y = -2x^2 + 5x - 2$ .

Пункт Таблица обеспечивает вывод таблицы значений аргумента и функции во введенном диапазоне.

Пункт меню Программа содержит две команды: Выход (которая закрывает форму) и О разработчике (которая выводит информацию об авторе проекта).

### 5 семестр

1. Парадигма функционального программирования.
2. Основные свойства функциональных языков.
3. Напишите функцию, которая по двум числам, представляющим собой коэффициенты уравнения  $ax + b = 0$ , возвращает пару первый элемент которой принимает значение истина, если уравнение имеет решение и ложь, в противном случае, а второй элемент является решением уравнения или равен 0, если решение не существует.
4. Напишите функцию, которая принимает значение истина, если два отрезка, концы которых  $(x_i, y_i)$ , где  $i = 1, 2, 3, 4$  параллельны или лежат на одной прямой.
5. Определите функцию, принимающую на вход целое число  $n$  и возвращающую список квадратов натуральных чисел, содержащий  $n$  элементов, упорядоченных по возрастанию.
6. Реализуйте на F# задачу вычисления числа  $P_i$  методом Монте-Карло
7. Постройте комбинатор *Next* со следующим свойством: для всякого комбинатора  $F$ , представляющего некоторую всюду определенную функцию  $f$ :, комбинатор  $(NextF)$  представляет функцию  $g(n) = f(n + 1)$ :

### Вопросы к зачету с оценкой по дисциплине (3 семестр)

1. Основы C++. Ввод-вывод информации.
2. Описание переменных в C++. Типы данных.
3. Оператор ветвления в полной и неполной форме.
4. Оператор выбора.
5. Циклы с предусловием и постусловием
6. Цикл `for`.
7. Массивы. Линейные массивы
8. Алгоритмы поиска и сортировок
9. Массивы. Двумерные массивы
10. Строки
11. Функции пользователя
12. Перегрузка функций
13. Структуры
14. Массивы структур
15. Исключения

### Вопросы к зачету по дисциплине (4 семестр)

1. Введение в Visual Studio.NET, Frame.NET. Язык программирования C#.
2. Ввод-вывод информации в C#
3. Понятия класса и объекта. Конструкторы и деструкторы класса
4. Методы класса.
5. Управление доступом к членам класса
6. Перегрузка методов и конструкторов
7. Визуальное программирование. Объекты, свойства, методы и события.
8. Визуальное программирование. Компонент Button/
9. Визуальное программирование. Компоненты Label, TextBox
10. Визуальное программирование. Компоненты CheckBox, RadioButton
11. Создание меню приложения. Компоненты MenuStrip, ContextMenuStrip
12. Визуальное программирование. Компонент Chart
13. Графические возможности среды. Компонент PictureBox

#### 14. Анимация в среде программирования C#

#### Вопросы к экзамену по дисциплине (5 семестр)

1. Парадигмы программирования
2. Языки функционального программирования
3. Язык программирования F#.
4. Анонимные функции. Сопоставление с образцом
5. Рекурсия
6. Цикл с аккумулятором
7. Списки.
8. Операции со списками
9. Переменные в языке F#.
10. Массивы в F#
11. Двоичные деревья в F#
12. Λ-исчисление

### 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

#### а) основная литература

1. Подбельский, В.В. Язык C#. Базовый курс: учеб. пособие / В.В. Подбельский. – М.: Финансы и статистика, 2015. – 408 с.  
<http://www.studentlibrary.ru/book/ISBN9785279035342.html>
2. Подбельский, В.В. Язык Си#. Решение задач [Электронный ресурс]: учеб. пособие / В.В. Подбельский. – М.: Финансы и статистика, 2014. – 296 с.  
<http://www.studentlibrary.ru/book/ISBN9785279035533.html>
3. Гуриков, С.Р. Введение в программирование на языке Visual C#: Учебное пособие / С.Р. Гуриков. – М.: Форум: НИЦ ИНФРА-М, 2013. – 448 с.  
<http://znanium.com/catalog.php?bookinfo=404441>
4. Агапов, В.П. Основы программирования на языке C# [Электронный ресурс]: учебное пособие/ В.П. Агапов – Электрон. текстовые данные. – М.: Московский государственный строительный университет, ЭБС АСВ, 2012. – 128 с.  
<http://www.iprbookshop.ru/16366>

#### б) дополнительная литература:

1. Фарафонов А.С. Программирование на языке высокого уровня [Электронный ресурс]: методические указания к проведению лабораторных работ по курсу «Программирование»/ Фарафонов А.С. – Электрон. текстовые данные. – Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2013. – 32 с.  
<http://www.iprbookshop.ru/22912>
2. Биллиг, В.А. Основы объектного программирования на C# (C# 3.0, Visual Studio 2008) [Электронный ресурс] / В.А. Биллиг. – Электрон. текстовые данные. – М.: БИНОМ. Лаборатория знаний, Интернет-Университет Информационных Технологий (ИНТУИТ), 2010. – 582 с.  
<http://www.iprbookshop.ru/16092>
3. Давыдова Н.А. Программирование [Электронный ресурс]: учебное пособие/ Давыдова Н.А., Боровская Е.В. – Электрон. текстовые данные. – М.: БИНОМ. Лаборатория знаний, 2015. – 239 с.  
<http://www.iprbookshop.ru/6485>

4. Степанов, В.П. Лабораторный практикум по программированию на языке Си: учеб.-метод. пособие по курсу "Информатика" [Электронный ресурс] / В.П. Степанов. – М.: Издательство МГТУ им. Н.Э. Баумана, 2007. –48 с.  
[http://www.studentlibrary.ru/book/bauman\\_0404.html](http://www.studentlibrary.ru/book/bauman_0404.html)
5. Бабушкина, И.А. Практикум по объектно-ориентированному программированию [Электронный ресурс]/ И.А. Бабушкина, С.М. Окулов – Электрон. текстовые данные. – М.: БИНОМ. Лаборатория знаний, 2015. – 367 с.  
<http://www.iprbookshop.ru/12254>
6. Окулов, С.М. Программирование в алгоритмах [Электронный ресурс] / С.М. Окулов. – М.: БИНОМ, 2014. – 383 с.: ил. – (Развитие интеллекта школьников).  
<http://www.studentlibrary.ru/book/ISBN9785996323111.html>

в) интернет-ресурсы:

1. [msdn.microsoft.com](http://msdn.microsoft.com)
2. <https://projecteuler.net/>
3. <http://fsharp.org/>

## **8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Проведение практических и лекционных занятий осуществляется с использованием мультимедийного комплекса (компьютер +проектор) или интерактивной доски.

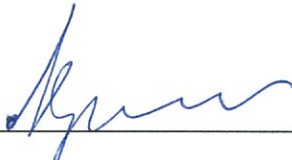
Лабораторные занятия проводятся в компьютерном классе с возможностью индивидуальной работы каждого студента за компьютером. Необходимое программное обеспечение – Microsoft Visual Studio 2008 (и выше).

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 44.03.05 Педагогическое образование.

Рабочую программу составил  
ст. преподаватель кафедры информатики и ИТО  А.А. Мартынова

Рецензент  
Заместитель директора  
ГБПОУ ВО «Владимирский педагогический колледж»  Н.И. Коршунова

Программа рассмотрена и одобрена на заседании кафедры информатики и ИТО  
протокол № 7а от 10.03.2016 года

Заведующий кафедрой  
информатики и ИТО  Ю.А. Медведев

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии  
направления 44.03.05 Педагогическое образование

Протокол № 3 от 17.03.16 года  
Председатель комиссии  М.В. Артамонова