

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**

УТВЕРЖДАЮ

Декан ФВТ



Л.Р. Фионова

« 30 » \_\_\_\_\_ 2017 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**С1.2.3 Технологии программирования**

Специальность – 09.05.01 *«Применение и эксплуатация автоматизированных систем специального назначения»*

Специализация №12: *«Автоматизированные системы обработки информации и управления специального назначения»*

Квалификация (степень) выпускника – *инженер*

Форма обучения – *очная*

Пенза, 2017

## 1. Цели освоения дисциплины

Целями освоения дисциплины «Технологии программирования» являются:

- приобретение студентами знаний о концепциях, моделях и принципах организации, положенных в основу "классических" технологий программирования и современных семейств технологий, о перспективных направлениях развития технологий,
- овладение студентами знаниями и навыками применения технологий проектирования и разработки программных продуктов,
- подготовка студентов к самостоятельному использованию современных инструментальных средств проектирования и разработки программного продукта,
- овладение студентами знаниями и навыками подготовки технической документации сопровождающих разработку программного обеспечения.

## 2. Место дисциплины в структуре ОПОП специалитета

Дисциплина «Технологии программирования» относится к базовой части С1.1 блока С1 ОПОП, шифр дисциплины С1.2.3.

Изучается дисциплина в 6-м семестре.

Изучение данной дисциплины базируется на следующих дисциплинах: "Математика", "Информатика", "Программирование", "Логика и основы алгоритмизации".

Основные положения дисциплины могут быть использованы при изучении следующих дисциплин: "Базы данных", "Защита информации", «Автоматизированные системы специального назначения», «Программирование в компьютерных сетях», а также при выполнении курсовых и дипломного проектов.

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины «Технологии программирования»

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данному направлению:

Коды компетенции	Наименование компетенции	Структурные элементы компетенции (в результате освоения дисциплины обучающийся должен знать, уметь, владеть)
ОПК-4	Способностью использовать языки и системы программирования, программные средства общего назначения, инструментальные средства компьютерного моделирования для решения различных исследовательских и профессиональных задач	<p><b>Знать:</b> языки и системы программирования общего назначения, инструментальные средства компьютерного моделирования; правила выполнения и чтения диаграмм, схем и другой рабочей проектной и технической документации; технологии программирования в интегрированной среде; методологии проектирования прикладных программ</p> <p><b>Уметь:</b> выполнять и читать диаграммы, схемы и другую рабочую проектную и техническую документацию; использовать основные подходы объектно-ориентированного программирования; отлаживать и тестировать программы с использованием инструментальных средств;</p>

		<p><b>Владеть:</b> навыками программирования в современных программных средах; основными технологиями проектирования программного обеспечения и технологиями программирования; владения основами коллективной разработки программных изделий; разработкой и отладкой программ.</p>
ПК-23	Способностью решать задачи анализа и синтеза элементов автоматизированных, систем специального назначения	<p><b>Знать:</b> стандарты оформления технической документации проектов;</p> <p><b>Уметь:</b> создания диаграмм, схем и сопровождающей документации</p>

#### 4. Структура и содержание дисциплины «Технологии программирования»

##### 4.1. Структура дисциплины

Общая трудоемкость дисциплины составляет 5 зачетных единиц, 180 часов.

№ п/п	Наименование разделов и тем дисциплины (модуля)	Семестр	Недели семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Формы текущего контроля успеваемости (по неделям семестра)	
				Аудиторная работа			Самостоятельная работа			Защита лабораторной работы	Курсовой проект
				Всего	Лекция	Лабораторные занятия	Всего	Подготовка к лабораторным работам	Курсовой проект		
1	Раздел 1. Программное обеспечение	6	1-2	4	4		4		4		
1.1	Тема 1.1. Технология программирования в историческом аспекте.	6	1	1	1						
1.2	Тема 1.2. Основные понятия и определения	6	1	1	1						
1.3	Тема 1.3. Классификация программного обеспечения		2	2	2						
2	Раздел 2. Технология разработки программных продуктов. Основные определения и подходы	6	3-6	12	8	4	12	8	4		
2.1	Тема 2.1 Особенности создания программного продукта		3	3	2	1				3	
2.2	Тема 2.2. Оценка качества процессов создания программного обеспечения.	6	4	3	2	1				4	
2.3	Тема 2.3. Жизненный цикл программ.	6	5-6	6	4	2				5	
3	Раздел 3. Проектирование программного обеспечения	6	7-13	54	14	40	50	20	30		
3.1	Тема 3.1. Проектирование программного обеспечения при структурном подходе.	6	7-8	14	4	10				8	
3.2	Тема 3.2. Проектирование программного	6	9-13	40	10	30				13	



## **4.2. Содержание дисциплины**

### **4.2.1.Содержание лекционного курса**

#### **Раздел 1. Программное обеспечение**

Тема 1.1. Технология программирования в историческом аспекте.  
Технологии программирования и основные этапы ее развития.

Тема 1.2. Основные понятия и определения.  
Термины и определения в технологиях программирования.

Тема 1.3. Классификация программного обеспечения.  
Системное программное обеспечение. Инструментарий технологии программирования. Пакеты прикладных программ

#### **Раздел 2. Технология разработки программных продуктов. Основные определения и подходы**

Тема 2.1. Особенности создания программного продукта.  
Проблематика проектирования. Эффективность и технологичность.

Тема 2.2. Оценка качества процессов создания программного обеспечения.  
Принципы работы с требованиями к программному обеспечению. Оценка стоимости ошибок. Управление требованиями. Последовательность работы с требованиями. Анализ проблемы.

Тема 2.3. Жизненный цикл программ.  
Каскадная модель, Модель с обратными связями. Спиралевидная модель.

#### **Раздел 3. Проектирование программного обеспечения**

Тема 3.1. Проектирование программного обеспечения при структурном подходе.  
Основа разработки программного обеспечения. Обзор средств проектирования программных продуктов

Тема 3.2. Проектирование программного обеспечения при объектном подходе.  
Определение ООП и его основные концепции. Особенности реализации. Основы UML для программистов.

#### **Раздел 4. Тестирование и отладка программ и сопровождение программ**

Тема 4.1. Виды тестирования.  
Ручной контроль программного обеспечения. Структурное тестирование. Функциональное тестирование. Комплексное тестирование.

Тема 4.2. Виды программных документов.  
Пояснительная записка. Руководство пользователя. Руководство системного программиста. Основные правила оформления текстовых документов.

#### **Раздел 5. Разработка программного обеспечения**

Тема 5.1. Инструментальные средства разработки программ.  
Назначение и виды инструментального ПО. Уровни инструментального ПО.

Тема 5.2. Технологии программирования.  
Гибкое проектирование программных продуктов. Экстремальное программирование.

#### 4.2.2. Перечень и содержание лабораторных занятий

№ п/п	№ раздела	Наименование лабораторных работ	Кол. часов
1	2	Разветвляющиеся вычислительные процессы	4
2	2	Стрельба по мишени	4
3	2	Организация циклов	6
4	3	Одномерные и многомерные массивы	8
5	3	Работа со строками	8
6	3	Классы и интерфейсы	18
7	4	Windows-приложение	6

#### 5. Образовательные технологии

В процессе изучения дисциплины применяются следующие образовательные технологии.

- Лекции с применением мультимедиа технологий
- Мастер-классы по работе с платформой Microsoft Visual Studio 2010.
- Разбор конкретных ситуаций при защите лабораторных работ.

В целях реализации индивидуального подхода к обучению студентов, осуществляющих учебный процесс по собственной траектории в рамках индивидуального рабочего плана, изучение данной дисциплины базируется на следующих возможностях: обеспечение внеаудиторной работы со студентами в том числе в электронной образовательной среде с использованием соответствующего программного обеспечения, дистанционных форм обучения, возможностей интернет-ресурсов, индивидуальных консультаций и т.д.

#### 6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

##### 6.1. План самостоятельной работы студентов

№ нед.	Тема	Вид самостоятельной работы	Задание	Рекомендуемая литература	Кол-во часов
1	Программное обеспечение	Проработка конспекта лекций и учебной литературы, подготовка к лабораторным работам.	Изучения основных понятий, связанных с разработкой программного обеспечения	/1/	4

3	Технология разработки программных продуктов. Основные определения и подходы	Проработка конспекта лекций и учебной литературы, подготовка к лабораторным работам.	Изучение особенностей создания программного обеспечения	/1/	12
7	Проектирование программного обеспечения	Проработка конспекта лекций и учебной литературы, подготовка к лабораторным работам.	Изучение структурного и объектного подходов разработки программного обеспечения. Методики выделения объектов из предметной области	/1,2/	50
14	Тестирование и отладка программ и сопровождение программ	Проработка конспекта лекций и учебной литературы, подготовка к лабораторным работам.	Изучение методов тестирования на этапах разработки программного обеспечения	/1,2,4/	12
17	Разработка программного обеспечения	Проработка конспекта лекций и учебной литературы, подготовка к лабораторным работам.	Изучение инструментария и гибких технологий разработки программного обеспечения	/1,2,3,4,5/	12

## 6.2. Методические указания по организации самостоятельной работы студентов

Планируются следующие виды самостоятельной работы:

- работа с лекционным материалом, предусматривающая проработку конспекта лекций и учебной литературы,
- подготовка к лабораторным работам,
- работа с конспектом лекций и изучение литературы при подготовке к зачету,
- курсовое проектирование.

## 6.3. Материалы для проведения текущего и промежуточного контроля знаний студентов

### *Контроль освоения компетенций*

№ п\п	Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются
1	Текущий:		



	Опрос перед каждой лекцией Защита лабораторных работ	Разделы 1 – 5	ОПК-4, ПК-23
2	Промежуточный: Защита курсового проекта	Разделы 1 – 5	ОПК-4, ПК-23
3	Промежуточный: зачет (2 вопроса и задача)	Разделы 1 – 5	ОПК-4, ПК-23

### *Темы курсовых проектов*

1. Программное приложение для анализа интерполяционных формул Лагранжа и Ньютона
2. Программное приложение для изучения свойств кривых Эрмита и кривых Безье
3. Программное приложение для анализа алгоритмов нахождения обратных матриц методами LU-разложения и матриц алгебраических дополнений
4. Программное приложение численного вычисления интегралов методами трапеций и прямоугольников
5. Программное приложение для анализа алгоритмов нахождения обратных матриц методами Гаусса-Жордана и матриц алгебраических дополнений
6. Программное приложение для изучения свойств кубических кривых и кривых Безье
7. Программное приложение “Нахождение корней уравнения методами дихотомии и касательных”
8. Программное приложение генерации лабиринта
9. Программное приложение “Исследование методов быстрой сортировки и Шелла”
10. Программное приложение для решения СЛАУ методами Гаусса и Крамера
11. Программное приложение “Исследование методов сортировок простыми вставками и быстрой сортировки”
12. Программное приложение для решения СЛАУ методами Гаусса и Гаусса-Зейделя
13. Программное приложение для изучения свойств B-сплайн кривых и кривых Безье
14. Программное приложение численного вычисления интегралов методами трапеций и прямоугольников
15. Программное приложение численного вычисления интегралов методами трапеций и прямоугольников
16. Программное приложение “Исследование методов сортировок простыми вставками и пузырьком”
17. Программное приложение “Исследование методов поиска перебором и бинарного поиска”
18. Программное приложение для решения СЛАУ методами Гаусса-Жордана и Гаусса-Зейделя
19. Программное приложение для изучения свойств кубических кривых и кривых Эрмита
20. Программное приложение для построения разверток геометрических 3d - фигур
21. Программное приложение “Нахождение корней уравнения методами дихотомии и касательных”
22. Программное приложение “Нахождение корней уравнения методами дихотомии и хорд”

23. Программное приложение “Исследование методов сортировок простыми вставками и Шелла”
24. Программное приложение “Исследование методов сортировок выбором и пузырьком”

*Демонстрационный вариант теста*

1. Что будет выведено в результате выполнения программы?

```
class Program
{
    static void Main(string[] args)
    {
        bool? b1 = true;
        bool b2 = false;
        b2 = b1;
        Console.WriteLine(b2);
        Console.ReadLine();
    }
}
```

- a) True  
 б) False  
 в) Возникнет ошибка на этапе компиляции.

2. Что будет выведено в результате выполнения программы?

```
class Program
{
    enum En
    {
        First = 15,
        Second,
        Third = 54
    }

    static void Main(string[] args)
    {
        Console.WriteLine((int)En.Second);
        Console.ReadLine();
    }
}
```

- a) 0  
 б) 1.  
 в) 16.  
 г) Возникнет ошибка на этапе компиляции.

3. Когда и почему использовать StringBuilder предпочтительнее чем string?

- a) Если строка редко изменяется.  
 б) Если строка часто изменяется  
 в) Если строка содержит спецсимволы.  
 г) Если строка содержит исключительно цифры

4. Что будет выведено в результате выполнения программы?

```
class Program
{
    static void Main(string[] args)
    {
        string s;
        Console.Write(Sum(5, 3, out s) + " ");
        Console.Write(s);
        Console.ReadLine();
    }

    static int Sum(int a, int b, out string s)
    {
        s = "abc";
        return a + b;
    }
}

```

a) 8 abc  
 б) 8  
 в) 8 8abc  
 г) Возникнет ошибка на этапе компиляции.

5. Для чего нужна конструкция try...catch ?

- a) "Отловить" ошибку
- б) Используется для ввода данных
- в) Используется для верных расчетов
- г) Обработать исключительную ситуацию

### ***Примерный перечень вопросов и заданий к зачету***

1. Технология программирования в историческом аспекте.
2. Основные понятия и определения дисциплины.
3. Системное программное обеспечение.
4. Инструментарий технологии программирования.
5. Пакеты прикладных программ.
6. Принципы работы с требованиями к программному обеспечению. Проблематика проектирования.
7. Оценка стоимости ошибок.
8. Управление требованиями.
9. Последовательность работы с требованиями. Анализ проблемы.
10. Преграды на пути выявления требований.
11. Понятие технологии разработки программы.
12. Основа разработки программного обеспечения.
13. Модели жизненного цикла.
14. Специфицирование и планирование в жизненном цикле.
15. Процесс разработки в жизненном цикле.
16. Выпуск продукта и механизмы обратной связи в жизненном цикле.
17. Определение требований к программным продуктам.
18. Функциональные требования к программным продуктам.
19. Эксплуатационные требования к программным продуктам.
20. Выбор архитектуры программного обеспечения.

21. Классификация структур данных.
22. Простые структуры данных.
23. Статические структуры данных.
24. Полустатические структуры данных.
25. Динамические структуры данных.
26. Понятие модуля.
27. Основные характеристики программного модуля.
28. Модульная структура программных продуктов.
29. Методы разработки при модульном программировании.
30. Спецификации процессов при структурном подходе.
31. Структурная схема разрабатываемого программного обеспечения.
32. Функциональная схема программного обеспечения.
33. Метод пошаговой детализации при составлении алгоритмов.
34. CASE-технологии.
35. Разработка структуры программного обеспечения при объектном подходе.
36. Основополагающие практики экстремального программирования.
37. Преимущества простого дизайна в экстремальном программировании.
38. Дизайн в экстремальном программировании.
39. Рефакторинг и принцип YAGNI.
40. Экстремальное программирование.
41. Суть проектирования. Программирование и тестирование.
42. Тестирование «белого ящика» и «черного ящика».
43. Порядок разработки тестов.
44. Автоматизация тестирования.
45. Модульное тестирование.
46. Интеграционное тестирование.
47. Системное тестирование.
48. Эффективность и оптимизация программ.
49. Стил программирования.
50. Методы оценки и измерения характеристик надежности.
51. Преимущества парного программирования.
52. Отладка программ.
53. Виды программных документов.
54. Выбор среды программирования.
55. Объектно-ориентированное программирование.
56. Основные сведения о защите программных продуктов.

### ***Пример задачи к зачету***

Разработать программное приложение, позволяющее выполнять математические операции над матрицами (сложение, вычитание, умножение, деление). При разработке использовать объектно-ориентированный подход.

## **7. Учебно-методическое и информационное обеспечение дисциплины «Разработка экономических приложений»**

### **а) основная литература:**

1. Иванова Г.С. Технология программирования : учебник / Г. С. Иванова. - 2-е изд., перераб. и доп. - М. : Изд-во МГТУ им.Н.Э.Баумана, 2006. - 336 с. (17 экз.)
2. Камаев В.А. Технологии программирования: учебник / В. А. Камаев, В. В. Костерин. - 2-е изд., перераб. и доп. - М. : Высш. шк., 2006. - 454 с. (20 экз.)

б) дополнительная литература:

3. Технологии программирования на языке C# в среде Microsoft Visual Studio 2013 Community: учеб.-метод.пособие / сост.А.П. Зимин. – Пенза: Изд-во ПГУ, 2016. – 68 с.

в) Интернет-ресурсы:

4. Сайт «Национальный Открытый Университет ИНТУИТ» Программирование на языке высокого уровня C#, URL: <http://www.intuit.ru/studies/courses/629/485/info>

г) Программное обеспечение:

5. Microsoft Visual Studio 2015 Community. <https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2015Community.aspx>.

### **8. Материально-техническое обеспечение дисциплины**

Лекционные занятия проводятся в аудитории, оснащенной ноутбуком, компьютерным проектором, проекционным экраном, шторами, сетью электропитания 220В.

Лабораторные занятия проводятся в классе, оснащенный персональными компьютерами и с установленным программным продуктом Microsoft Visual Studio 2010.

Рабочая программа дисциплины «Технологии программирования» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения».

Программу составил:

Зимин А. П., ст. преподаватель кафедры  
«Информационно-вычислительные системы»



**Настоящая программа не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.**

Программа одобрена на заседании кафедры «Информационно-вычислительные системы»

Протокол № 14

от «27» 06 2017 года

Зав. кафедрой ИВС



Косников Ю.П.

Программа одобрена методической комиссией факультета вычислительной техники

Протокол № 9

от «30» 06 2017 года

Председатель методической комиссии ФВТ



Глотова Т. В.

**Сведения о переутверждении программы на очередной учебный год и  
регистрации изменений**

Учебный год	Решение кафедры (№ протокола, дата, подпись зав. кафедрой)	Внесенные изменения	Номера листов (страниц)		
			заменен- ных	новых	аннулиро- ванных