

**Филиал федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Национальный исследовательский университет «МЭИ»  
в г. Смоленске**

**УТВЕРЖДАЮ**

Зам. директора  
филиала ФГБОУ ВО «НИУ «МЭИ»  
в г. Смоленске  
по учебно-методической работе  
В.В. Рожков  
« 31 » 08 2015 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ**

(НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ)

**Направление подготовки: 09.03.01 Информатика и вычислительная техника**

**Профиль подготовки: Вычислительные машины, комплексы, системы и сети**

**Уровень высшего образования: бакалавриат**

**Нормативный срок обучения: 4 года**

**Форма обучения: очная**

**Смоленск – 2015 г.**

## **1 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы**

Курс «Технология программирования» является вводным курсом по структурному и объектно-ориентированному проектированию для студентов направления подготовки 09.03.01.

**Целью освоения дисциплины** «Технология программирования» является обучение студентов систематизированному представлению о принципах построения и проектирования программных систем.

**Задачами дисциплины являются:**

- познакомить обучающихся с основными понятиями и определениями, с классификацией программного обеспечения;
- дать представление об этапах создания программного продукта в рамках жизненного цикла, о современном состоянии технологий разработки программного продукта;
- познакомить обучающихся с существующими подходами к оценке качества процессов создания программного обеспечения, ;
- дать обучающемуся практические навыки проектирования программного обеспечения и расчета его надежности.

Дисциплина направлена на формирование следующих общекультурных и профессиональных компетенций:

- ОК-6. Стремится к саморазвитию, повышению своей квалификации и мастерства
- ОК-8. Осознает социальную значимость своей будущей профессии, обладает высокой мотивацией к выполнению профессиональной деятельности
- ОК-12. Имеет навыки работы с компьютером как средством управления информацией;
- ОК-13. Способен работать с информацией в глобальных компьютерных сетях;
- ПК-2. Осваивать методики использования программных средств для решения практических задач
- ПК-3. Разрабатывать интерфейсы "человек - электронно-вычислительная машина"
- ПК-4. Разрабатывать модели компонентов информационных систем, включая модели баз данных
- ПК-5. Проектно-технологическая деятельность: разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования
- ПК-11. сервисно-эксплуатационная деятельность: устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем

В результате изучения дисциплины обучаемый должен:

**знать:**

- принципы проектирования программных систем (ОК-6);
- организацию процесса проектирования программного обеспечения (ОК-12, ПК-3);
- методологию структурного проектирования ПО (ОК-13, ПК-4, ПК-5);
- методологию объектно-ориентированного проектирования ПО (ОК-13, ПК-4, ПК-5);
- технологические средства разработки программного обеспечения (ОК-8);
- методы декомпозиции и абстракции при проектировании ПО (ПК-2);
- методы отладки и тестирования программ (ПК-11);
- методы защиты программ и данных (ОК-13).

**уметь:**

- использовать методы декомпозиции и абстракции при проектировании ПО (ОК-6, ПК-2);

- применять средства разработки программного обеспечения: инструментальные среды разработки, средства поддержки проекта, отладчики (ОК-12);
- документировать и оценивать качество программных продуктов (ПК-4, ПК-5, ПК-11);
- проектировать пользовательские интерфейсы (ОК-8, ОК-13, ПК-3).

**владеть:**

- методами и средствами разработки и оформления технической документации (ОК-12, ПК-4, ПК-5, ПК-11);
- методами проектирования программного обеспечения при структурном и объектно-ориентированном подходе (ПК-2, ПК-3, ПК-4, ПК-5);
- методами структурного и функционального тестирования (ПК-2, ПК-11)
- методами совместной разработки приложений (ОК-6, ОК-8, ОК-13, ПК-5).

**2 Место дисциплины в структуре ООП ВПО направления 09.03.01 бакалавр по направлению подготовки «Информатика и вычислительная техника»**

Дисциплина «Технология программирования» относится к обязательным дисциплинам базовой части профессионального цикла Б3.В.ОД.6 основной образовательной программы подготовки бакалавров по направлению «09.03.01 «Информатика и вычислительная техника».

В соответствии с учебным планом по направлению "Информатика и вычислительная техника" дисциплина «Вычислительные системы» базируется на следующих дисциплинах:

- Б1.Б.3 Философия
- Б1.В.ДВ.1.1 Психологические основы профессиональной деятельности
- Б2.Б.1.1 Алгебра и геометрия
- Б2.Б.1.2 Математический анализ
- Б2.Б.2 Физика
- Б2.Б.3 Информатика
- Б2.В.ОД.1 Математическая логика и теория алгоритмов
- Б2.В.ОД.2 Дискретная математика
- Б3.Б.3 Операционные системы
- Б3.Б.9.1 ЭВМ
- Б5.У.1 Учебная практика
- Б1.Б.3 Философия
- Б1.В.ДВ.1.1 Психологические основы профессиональной деятельности
- Б2.Б.1.1 Алгебра и геометрия
- Б2.Б.1.2 Математический анализ
- Б2.Б.2 Физика
- Б2.Б.3 Информатика
- Б2.В.ОД.1 Математическая логика и теория алгоритмов
- Б2.В.ОД.2 Дискретная математика
- Б3.Б.3 Операционные системы

Знания, умения и навыки, полученные студентами в процессе изучения дисциплины «Технология программирования» необходимы для формирования компетенций в дисциплинах:

- Б2.Б.4 Экология
- Б2.В.ДВ.1.1 Теория принятия решений
- Б2.В.ДВ.1.2 Исследование операций

- Б2.В.ОД.5 Прикладная статистика
- Б3.Б.1.2 Схемотехника
- Б3.Б.10 Метрология, стандартизация и сертификация
- Б3.Б.5 Сети и телекоммуникации
- Б3.Б.7 Базы данных
- Б3.Б.8 Защита информации
- Б3.Б.9.2 Периферийные устройства
- Б3.В.ДВ.1.1 Основы логического программирования
- Б3.В.ДВ.1.2 Кластерные вычислительные системы
- Б3.В.ДВ.2.1 Инженерное проектирование и САПР
- Б3.В.ДВ.2.2 Лингвистическое и программное обеспечение САПР
- Б3.В.ДВ.3.1 Теория автоматов
- Б3.В.ДВ.3.2 Аппаратные и программные средства
- Б3.В.ДВ.4.1 Структурный анализ и проектирование информационных систем
- Б3.В.ДВ.4.2 Информационные технологии
- Б3.В.ДВ.5.1 Технология объектного программирования
- Б3.В.ДВ.5.2 Вычислительные системы
- Б3.В.ОД.2 Моделирование
- Б3.В.ОД.3 Основы теории управления
- Б3.В.ОД.4 Микропроцессорные системы
- Б3.В.ОД.5 Системное программное обеспечение
- Б3.В.ОД.7 Электронные цепи ЭВМ

**3 Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся)**

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 часа.

Объем занятий, проводимых в интерактивной форме, 16 часов

### Аудиторная работа

| Цикл:   | Б3 Профессиональный цикл                  | Семестр   |
|---|---|-----------|
| Часть цикла:  | Вариативная<br>Обязательная<br>дисциплина |           |
| Индекс дисциплины по учебному плану                               | Б3.В.ОД.6                                 |           |
| Часов всего по учебному плану                                     | 144                                       | 4 семестр |
| Трудоемкость в зачетных единицах (ЗЕТ)                            | 4   | 4 семестр |
| Лекции (ЗЕТ/ часов)   | 1/36                                      | 4 семестр |
| Практические занятия (ЗЕТ/ часов)                                 | 1/36                                      | 4 семестр |
| Объем самостоятельной работы по учебному плану (ЗЕТ/ часов всего) | 1/36                                      | 4 семестр |
| Зачет с оценкой (в объеме самостоятельной работы)                 | -   | 4 семестр |
| Экзамен   | 1/36                                      | 4 семестр |

### Самостоятельная работа студента

| Вид работ   | Трудоёмкость |           |
|---|--------------|-----------|
|   | ЗЕТ          | час       |
| Подготовка к сеансу тестирования                          | 0,1          | 3         |
| Подготовка к контрольной работе                           | 0,2          | 8         |
| Подготовка к выполнению и защите лабораторных работ (лаб) | 0,64         | 23        |
| Изучение дополнительного теоретического материала         | 0,06         | 2         |
| <b>Всего:</b>   | <b>1</b>     | <b>36</b> |

### Распределение трудоемкости дисциплины по семестрам и видам учебной работы

| № п/п | Разделы и темы дисциплины  | Семестр | Неделя семестра | Общая трудоемкость, всего | Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах) |        |                      |                     |         |                        |              |                        |        | Занятия в интерактивной форме | Формы текущего контроля успеваемости (по каждой теме)<br>Форма промежуточной аттестации (по семестрам: зачет с оценкой, экзамен) |
|-------|--|---------|-----------------|---------------------------|--|--------|----------------------|---------------------|---------|------------------------|--------------|------------------------|--------|-------------------------------|--|
|       |  |         |                 |                           | Аудиторные занятия   |        |                      |                     | Экзамен | Самостоятельная работа |              |                        |        |                               |  |
|       |  |         |                 |                           | Всего  | Лекции | Практические занятия | Лабораторные работы |         | Всего                  | Реферат, РГР | Курсовая работа/проект | Другая |                               |  |
| 1     | Основы программирования в среде C#   | 4       | 1               | 22                        | 12   | 4      |                      | 8                   | 4       | 6                      |              |                        | 6      | 2                             | Тест   |
| 2     | Основные понятия и подходы. Приемы обеспечения технологичности и программных продуктов | 4       | 3               | 15                        | 8  | 4      |                      | 4                   | 2       | 5                      |              |                        | 5      | 1                             | Контр. Работа  |
| 3     | Разработка технического задания  | 4       | 5               | 6                         | 2  | 2      |                      |                     | 2       | 2                      |              |                        | 2      | 1                             | Контр. Работа  |
| 4     | Проектирование программного обеспечения  | 4       | 6               | 22                        | 10   | 6      |                      | 4                   | 8       | 4                      |              |                        | 4      | 3                             | Устный опрос   |

|              |   |              |    |            |           |           |  |           |           |           |  |  |           |            |               |
|--------------|---|--------------|----|------------|-----------|-----------|--|-----------|-----------|-----------|--|--|-----------|------------|---------------|
|              | при структурном подходе к программированию  |              |    |            |           |           |  |           |           |           |  |  |           |            |               |
| 5            | Тестирование и отладка программных продуктов при структурном подходе к программированию         | 4            | 9  | 26         | 14        | 6         |  | 8         | 6         | 6         |  |  | 6         | 3          | Контр. Работа |
| 6            | Проектирование программного обеспечения при объектно-ориентированном подходе к программированию | 4            | 12 | 25         | 12        | 8         |  | 4         | 8         | 5         |  |  | 5         | 3          | Контр. Работа |
| 7            | Разработка пользовательских интерфейсов   | 4            | 16 | 18         | 8         | 4         |  | 4         | 4         | 6         |  |  | 6         | 1          | Устный опрос  |
| 8            | Оценка качества программного обеспечения  | 4            | 18 | 10         | 6         | 2         |  | 4         | 2         | 2         |  |  | 2         | 2          | Устный опрос  |
| <b>Всего</b> |   | <b>часов</b> |    | <b>144</b> | <b>72</b> | <b>36</b> |  | <b>36</b> | <b>36</b> | <b>36</b> |  |  | <b>36</b> | <b>16</b>  |               |
|              |   | <b>ЗЕТ</b>   |    | <b>4</b>   | <b>2</b>  | <b>1</b>  |  | <b>1</b>  | <b>1</b>  | <b>1</b>  |  |  | <b>1</b>  | <b>0,4</b> |               |

**Матрица соотнесения тем/разделов дисциплины и формируемых в них общекультурных и профессиональных компетенций**

| Темы, разделы дисциплины | Количество часов | Код компетенции |          |          |          |          |          |          |          |          |     |    | Σ общее количество компетенций |  |           |
|--------------------------|------------------|-----------------|----------|----------|----------|----------|----------|----------|----------|----------|-----|----|--------------------------------|--|-----------|
|                          |                  | ОК-6            | ОК-8     | ОК-12    | ОК-13    | ПК-2     | ПК-3     | ПК-4     | ПК-5     | ПК-11    | ... | .. |                                |  |           |
| Тема 1                   | 20               | X               |          | X        |          |          |          |          |          |          |     |    |                                |  | 2         |
| Тема 2                   | 12               |                 |          |          |          |          |          |          |          |          | X   |    |                                |  | 1         |
| Тема 3                   | 10               |                 |          |          |          |          |          |          |          | X        |     |    |                                |  | 1         |
| Тема 4                   | 25               |                 |          |          |          | X        |          |          | X        |          |     |    |                                |  | 2         |
| Тема 5                   | 23               |                 |          |          |          | X        |          |          | X        |          |     |    |                                |  | 2         |
| Тема 6                   | 34               |                 |          |          |          | X        |          | X        | X        |          |     |    |                                |  | 3         |
| Тема 7                   | 14               |                 |          |          | X        |          | X        |          |          |          |     |    |                                |  | 2         |
| Тема 8                   | 6                |                 | X        |          |          |          |          |          |          |          |     |    |                                |  | 1         |
| <b>Итого</b>             | <b>144</b>       | <b>1</b>        | <b>1</b> | <b>1</b> | <b>1</b> | <b>3</b> | <b>1</b> | <b>1</b> | <b>4</b> | <b>1</b> |     |    |                                |  | <b>14</b> |

**4 Содержание дисциплины, структурированное по темам с указанием отведенного на них количества академических часов и видов учебных занятий**

| №№ пп | Наименование раздела, темы дисциплины  | Содержание  | Коды формируемых компетенций | Результаты освоения  |
|-------|--|---|------------------------------|--|
| 1.    | Тема 1. Основы программирования в среде C#   | Язык программирования C#. Структура программы. Операции уменьшения, увеличения, операции условия. Логические операции. Типы данных. Приведение типов. Массивы. Операторы условия и выбора. Операторы циклов. Классы для работы со строками. Регулярные выражения. Классы в C#. Интерфейсы, делегаты, события.                           | ОК-6, ОК -12                 | ОК-6: Освоение среды программирования Visual Studio – создание проектов, тестирование, отладка   |
| 2     | Тема 2. Основные понятия и подходы. Приемы обеспечения технологичности и программных продуктов | Этапы развития технологии программирования. Модели жизненного цикла программного обеспечения. Понятия эффективности и технологичности программного обеспечения. Модули и их свойства Средства описания структурных алгоритмов (псевдокоды, блок-схемы алгоритмов, Flow-формы, диаграммы Насси-Шнейдермана). Правила оформления программ | ПК -11                       | ПК-11: Разработка технологичных много модульных программных продуктов с учетом сцепления и связности модулей, освоение методов проектирования структурных алгоритмов |
| 3     | Тема 3. Разработка технического задания  | Разработка технического задания<br>Классификация моделей разрабатываемого программного обеспечения  | ПК -5                        | ПК-5: освоение методики разработки технического задания, методов предпроектных исследований, осознанный выбор среды программирования                                 |
| 4     | Тема 4. Проектирование программного обеспечения при структурном                                | Диаграммы переходов состояний. Функциональные диаграммы. Диаграммы потоков данных. Диаграммы отношений компонентов данных: диаграммы Джексона   | ПК-2, ПК-5                   | ПК-2: Осваивать методики использования программных средств для решения практических задач  |

| №№<br>пп | Наименование<br>раздела, темы<br>дисциплины   | Содержание   | Коды<br>форми-<br>руемых<br>компе-<br>тенций | Результаты освоения  |
|----------|---|--|--|--|
|          | подходе к<br>программирова<br>нию   | и скобочные диаграммы Орра,<br>сетевая модель данных<br>(Диаграммы «сущность-<br>связь»). Структурная и<br>функциональная схемы.<br>Структурные карты<br>Константайна  |  | ПК-5:<br>Освоить методику<br>структурного<br>проектирования SADT,<br>разработку диаграмм<br>переходов состояний,<br>структурных и<br>функциональных схем,<br>методику Джексона,<br>Структурные карты<br>Константайна   |
| 5        | Тема 5.<br>Тестирование и<br>отладка<br>программных<br>продуктов при<br>структурном<br>подходе                                      | Структурное тестирование.<br>Функциональное<br>тестирование. Классификация<br>ошибок. Методы отладки<br>программного обеспечения   | ПК-2, ПК-5                                   | ПК-2: Освоить методы<br>ручного, структурного<br>и функционального<br>тестирования   |
| 6        | Тема 6.<br>Проектировани<br>е программного<br>обеспечения<br>при объектно-<br>ориентированн<br>ом подходе к<br>программирова<br>нию | UML- стандартный язык<br>описания разработки<br>программных продуктов с<br>использованием объектного<br>подхода. Диаграммы<br>вариантов использования.<br>Диаграмма классов.<br>Диаграмма<br>последовательностей.<br>Диаграмма деятельности.<br>Диаграмма пакетов.<br>Диаграмма кооперации.<br>Диаграммы состояний<br>объекта. Диаграмма<br>компонентов. Диаграмма<br>размещения | ПК-2, ПК-4,<br>ПК-5                          | ПК-2: Осваивать<br>методику<br>использования<br>программных средств<br>для решения<br>практических задач<br>ПК-5 разрабатывать<br>компоненты<br>программных<br>комплексов и баз<br>данных, использовать<br>современные<br>инструментальные<br>средства и технологии<br>программирования.<br>ПК-4: Освоить язык<br>UML для<br>проектирования<br>объектно-<br>ориентированных<br>программных<br>комплексов |



| №№<br>пп | Наименование<br>раздела, темы<br>дисциплины                  | Содержание  | Коды<br>форми-<br>руемых<br>компе-<br>тенций | Результаты освоения  |
|----------|--|---|--|--|
| 7        | Тема 7.<br>Разработка<br>пользовательских<br>интерфейсов     | Разработка пользовательского<br>интерфейса. Граф диалога с<br>пользователем | ОК-13, ПК-<br>3                              | ОК-13: Способен<br>работать с<br>информацией в<br>глобальных<br>компьютерных сетях<br>ПК3: Разрабатывать<br>интерфейсы "человек -<br>электронно-<br>вычислительная<br>машина". Разработка<br>много модульных<br>многооконных<br>приложений, освоение<br>методов<br>межпроцессных<br>взаимодействий.<br>Использование при<br>проектировании графа<br>диалога с<br>пользователем |
| 8        | Тема 8.<br>Оценка<br>качества<br>программного<br>обеспечения | Оценка качества<br>программного обеспечения по<br>ГОСТ 28195-89             | ОК-8   | ОК-8: Выполнять<br>оценку качества<br>разработанных<br>программных<br>продуктов  |

Лекционные занятия (в количестве 36 часов) проводятся в интерактивной форме (используются технологии типа «лекция-провокация», т.е. в процессе лекции делается преднамеренная ошибка с последующим опросом студентов на следующей лекции и организацией диалога «преподаватель-студент», «студент-студент» с целью выявления ошибки и установления истины.

#### Лабораторные работы

| №<br>семес-<br>тра | Раздел учебной<br>дисциплины                     | Наименование практических занятий                        | Трудоем-<br>кость,<br>часы |
|--------------------|--|--|----------------------------|
| 4                  | Тема 1. Основы<br>программирования в<br>среде C# | 1. Линейные программы<br>2. Циклические программы        | 2<br>2                     |
| 4                  | Тема 1. Основы<br>программирования в<br>среде C# | 1. Одномерные массивы<br>2. Двумерные массивы<br>3. Тест | 1<br>2<br>1                |

|   |   |  |                        |
|---|---|--|------------------------|
|   | Тема 2. Основные понятия и подходы. Приемы обеспечения технологичности программных продуктов            | 1. Работа со строками класса String<br>2. Работа со строками класса StringBuilder<br>3. Регулярные выражения<br>4. Оценка характеристик разработанных программ с помощью метрик Холстеда; оценка характеристик разработанных программ с помощью метрик Джилба; оценка надежности программных средств с помощью модели Джелински – Моранды, эвристической модели и модели Нельсона; | 1<br>1<br>1<br>1       |
| 4 | Тема 4. Проектирование программного обеспечения при структурном подходе к программированию              | 1. Работа с файлами.<br>2. Оценка степени отлаженности разрабатываемых приложений с помощью модели Миллса  | 2<br>2                 |
| 4 | Тема 5. Тестирование и отладка программных продуктов при структурном подходе                            | 1. Выполнить ручное тестирование разработанных приложений<br>2. Выполнить структурное тестирование разработанных приложений<br>3. Выполнить функциональное тестирование разработанных приложений   | 1<br>2<br>1            |
| 4 | Тема 6. Проектирование программного обеспечения при объектно-ориентированном подходе к программированию | 1. Работа с классами в С#. Наследование<br>2. Разработка UML - диаграмм  | 2<br>2                 |
| 4 | Тема 7 . Разработка пользовательских интерфейсов<br>Тема 8. Оценка качества программного обеспечения    | 1. Разработать объектно-ориентированное, многооконное приложение с использованием интерфейсов, делегатов и событий.<br>2. Разработка UML – диаграмм<br>3. Обосновать тип пользовательского интерфейса и форму диалога<br>4. Разработать граф диалога пользователя<br>5. Разработать пользовательское меню  | 2<br>0,5<br>0,5<br>0,5 |
| 4 | Тема 8. Оценка качества программного обеспечения  | 1. Выполнить оценку качества разработанного приложения по ГОСТ 28195-89  | 4                      |

Лабораторные работы (36 часов) проводятся в интерактивной форме. Каждому студенту выдается индивидуальное задание. Затем организуется активный диалог студентов с преподавателем и между собой для подведения итогов решения задания и определения его практической значимости.

## 5 Самостоятельная работа студента

Для обеспечения самостоятельной работы разработаны:

1. Конспект лекций по дисциплине (см. приложение 3.РПД Б3.В.ОД.6 (лк));
2. Методические указания к выполнению лабораторных работ (см. приложение 3.РПД Б3.В.ОД.6 (лб));
3. Методические указания к самостоятельной работе студентов (см. приложение 3.РПД Б3.В.ОД.6 (срс)).

### Самостоятельная работа студентов по темам дисциплины в часах

| № п/п | № семестра | Раздел, тема учебной дисциплины | Вид самостоятельной работы студента (курсовой проект, курсовая работа, реферат, расчетно-графическая работа, др.) | Всего часов |
|-------|------------|---------------------------------|---|-------------|
| 1     | 4          | Тема 1.                         | Подготовка к сеансу тестирования  | 3           |
|       |            |                                 | Оформление и подготовка к защите лабораторной работы (2 лабораторные работы)                                      | 3           |
| 2     | 4          | Тема 2                          | Подготовка к контрольной работе   | 2           |
|       |            |                                 | Оформление и подготовка к защите лабораторных работ   | 3           |
| 3     | 4          | Тема 3                          | Подготовка к контрольной работе   | 2           |
| 4     | 4          | Тема 4                          | Оформление и подготовка к защите лабораторных работ,  | 4           |
| 5     | 4          | Тема 5                          | Оформление и подготовка к защите лабораторных работ   | 4           |
|       |            |                                 | Подготовка к контрольной работе   | 2           |
| 6     | 4          | Тема 6                          | Оформление и подготовка к защите лабораторной работы  | 3           |
|       |            |                                 | Подготовка к контрольной работе   | 2           |
| 7     | 4          | Тема 7                          | Оформление и подготовка к защите лабораторных работ   | 4           |
|       |            |                                 | Изучение дополнительного теоретического материала   | 2           |
| 8     | 4          | Тема 8                          | Оформление и подготовка к защите лабораторных работ   | 2           |
| 9     | 4          | ИТОГО                           |   | 36          |

Для обеспечения самостоятельной работы разработаны методические указания по самостоятельной работе при подготовке к лабораторным работам.

### Виды контроля самостоятельной работы студентов и оценочные средства

| № п/п | № семестра | Раздел, тема учебной дисциплины | Виды контроля      | Оценочные средства   |
|-------|------------|---------------------------------|--------------------|--|
| 1     | 4          | Тема 1                          | Тест               | «50%»-Пороговый уровень освоения компетенции<br>«70%»- Продвинутый уровень освоения компетенции<br>«90%»- Высокий уровень освоения компетенции |
| 2     | 4          | Тема 2,3,5,6                    | Контрольная работа | «3»- Пороговый уровень освоения компетенции<br>«4»- Продвинутый уровень освоения компетенции<br>«5»- Высокий уровень освоения компетенции      |

### Самостоятельная работа студентов по темам дисциплины

Лабораторные работы выполняются по методическому пособию:

Гаврилов А.И., Малахов В.В. Малашенкова И.В., Панкратова Е.А., Семенова О.В. Основы программирования в среде С#. Методические указания к лабораторным работам по дисциплине «Технологии программирования» – Смоленск: РИО филиала МЭИ в г.Смоленске, 2014. – 110 с.

#### Тема 1.

#### Лабораторная работа № 1

#### ЦЕЛЬ РАБОТЫ:

1. Изучение основ языка С#;
2. Знакомство с элементами управления С#.
3. Составление линейных и циклических программ.
4. Работа с математическими функциями С#.

#### Вопросы и задания для самостоятельной работы (Примеры вариантов):

#### Линейные программы

Напишите программу расчета по двум формулам

| Вариант | Формула 1   | Формула 2   |
|---------|---|---|
| 1       | $z_1 = \cos\alpha + \sin\alpha + \cos 3\alpha + \sin 3\alpha$                               | $z_2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5}{2}\pi - 8\alpha\right)$ |
| 2       | $z_1 = \cos\alpha + \sin\alpha + \cos 3\alpha + \sin 3\alpha$                               | $z_2 = 2\sqrt{2} \cos\alpha * \sin\left(\frac{\pi}{4} + 2\alpha\right)$     |
| 3       | $z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos\alpha + 1 - 2\sin^2 2\alpha}$ | $z_2 = 2\sin\alpha$   |

#### Циклические алгоритмы

Составить алгоритм и написать программу для вычисления суммы S заданного ряда с погрешностью не более  $\epsilon_{ps}$  при различных значениях аргумента X. Во внешнем цикле организовать изменение X от 0.5 до 0.75 с шагом 0.05. Во внутреннем осуществлять сложение членов ряда до тех пор, пока не будет достигнута заданная точность. Вычисление очередного члена ряда основывать на промежуточных результатах, полученных во время вычисления предыдущего члена ряда. Дополнительно для контроля выводить значение соответствующей функции при каждом значении X. Варианты задания приведены в таблице.

| № | Суммируемый ряд   | Функция                        |
|---|---|--------------------------------|
| 1 | $\frac{2x^2(3+x)}{3!} + \frac{2x^6(7+x)}{7!} + \dots + \frac{2x^{4k-2} \cdot (4k-1+x)}{(4k-1)!} + \dots$    | $e^x - \sin x - \cos x$        |
| 2 | $\frac{3x^3}{1!} + \frac{5x^5}{2!} + \frac{7x^7}{3} + \dots + \frac{2k+1}{k!} x^{2k+1} + \dots$             | $(x + 2x^3) \cdot e^{x^2} - x$ |
| 3 | $\frac{x}{3!} - \frac{x^3}{5!} + \frac{x^5}{7!} - \dots + (-1)^{k-1} \cdot \frac{x^{2k-1}}{(2k+1)} + \dots$ | $\frac{x - \sin x}{x^2}$       |

## Тема 1.

### Лабораторная работа № 2

#### Цель работы:

1. Изучить особенности работы с массивами в языке C#
2. Изучить свойства и методы класса System.Array.

#### Одномерные массивы (примеры заданий)

##### Вариант 1

В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить

- сумму отрицательных элементов массива;
- произведение элементов массива, расположенных между максимальным и минимальным элементами.
- Упорядочить элементы массива по возрастанию.

##### Вариант 2

В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить

- сумму положительных элементов массива;
- произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.
- Упорядочить элементы массива по убыванию.

##### Вариант 3

В одномерном массиве, состоящем из  $n$  целочисленных элементов, вычислить:

- произведение элементов массива с четными номерами;
- сумму элементов массива, расположенных между первым и последним нулевыми элементами.
- Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом — все отрицательные (элементы, равные нулю, считать положительными).

#### Двумерные «прямоугольные» массивы (примеры заданий)

##### Вариант 1

Дана целочисленная прямоугольная матрица. Определить:

- количество строк, не содержащих ни одного нулевого элемента;
- максимальное из чисел, встречающихся в заданной матрице более одного раза.

##### Вариант 2

Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента.

Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

##### Вариант 3

Дана целочисленная прямоугольная матрица. Определить:

- количество столбцов, содержащих хотя бы один нулевой элемент;
- номер строки, в которой находится самая длинная серия одинаковых элементов.

**Двумерные «ступенчатые массивы» (примеры заданий)**

Сформировать двумерный «ступенчатый» массив, имеющий вид, изображенный в таблице  
Таблица – Задания на двумерные «ступенчатые» массивы

| № | Задание | №  | Задание |
|---|---------|----|---------|
| 1 |         | 11 |         |
| 2 |         | 12 |         |
| 3 |         | 13 |         |

**Тема 2**

**Лабораторная работа № 3**

**Цель работы:** Изучение классов языка программирования C# для работы со строками:

1. Работа с функциями класса System.String
2. Работа с функциями класса System.Text.StringBuilder
3. Работа с регулярными выражениями. Поиск в тексте фрагментов по определенному шаблону
4. Оценить технологичность разработанного программного обеспечения (сцепление, связность, эффективность)
5. Оценка характеристик разработанных программ с помощью метрик Холстеда; оценка характеристик разработанных программ с помощью метрик Джилба; оценка надежности программных средств с помощью модели Джелински – Моранды, эвристической модели и модели Нельсона

**Работа со строками класса String и StringBuilder (примеры заданий)**

**Вариант 1**

1. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит каждое слово в обратном порядке.
2. Дана строка не более чем из шести произвольных различных символов. Разработать программу вывода всех возможных подстрок, составленных из символов данной строки

**Вариант 2**

1. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит на экран только предложения, содержащие введенное с клавиатуры слово.
2. По правилам машинописи после запятой в тексте всегда ставится пробел. Составить программу исправления такого типа ошибок в тексте.

#### Вариант 3

1. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит на экран только строки, содержащие двузначные числа.
2. Дан текст. Вывести слова, которые отличны от последнего слова и в них нет повторяющихся букв

#### Регулярные выражения (примеры заданий)

Используя регулярные выражения, напишите следующие приложения:

1. Дан текстовый файл  $f$  и строка  $s$ , Переписать в новый файл  $g$  все строки файла  $f$ , содержащие значения переменной  $s$  в качестве подстроки.
2. Дан текстовый файл. Вывести все слова, начинающиеся с согласных букв русского алфавита.
3. Дан текстовый файл. Вывести все слова, состоящие только из цифр.

#### Тема 4

##### Цель работы:

1. Рассмотреть возможности C# для работы с потоками байтов символов и строк.
2. Оценить степень отлаженности разрабатываемых приложений с помощью модели Миллса
3. Разработать структурную и функциональную схемы приложений
4. Построить диаграммы IDEF0

##### Примеры заданий

###### Вариант 1.

1. Сформировать файл последовательности 15 чисел, в которой каждый  $i$ -й компонент определяется по формуле

$$y = \sin(i \cdot \pi / 8), \quad \text{если } i \leq 8;$$
$$y = 4 \cos(i(\pi + 1)/5), \quad \text{если } i > 8.$$

Определить количество положительных значений, содержащихся в сформированном файле.

2. Составить программу, определяющую правильность следования скобок в строке символов, используя для этой цели стек на основе файла.

###### Вариант 2.

1. Сформировать файл последовательности 20 чисел, в которой каждый  $i$ -й компонент определяется по формуле

$$y = \sin(i \cdot \pi / 8), \quad \text{если } i \leq 8;$$
$$y = 4 \cos(i(\pi + 1)/5), \quad \text{если } i > 8.$$

Определить количество отрицательных значений, содержащихся в сформированном файле.

2. Подсчитать количество двоек символов 'ee', 'nn', 'll' в тексте, расположенном в текстовом файле.

###### Вариант 3.

1. Сформировать файл из значений случайных величин:

0,324; 0,524; 0,789; 0,556; 0,761; 0,248; 0,345; 0,911; 0,216,

Определить для данной последовательности среднее арифметическое компонентов, значения которых меньше 0,5.

2. Разбить произвольный текст, находящийся в файле, на строки определенной длины. При переносе слова предусмотреть вывод дефиса.

## Тема 5

**Цель работы:** Изучение методов тестирования программного обеспечения:

- 1) Выполнение ручного тестирования программ
- 2) Выполнение структурного тестирования программ следующими методами:
  - a. тестирование базового пути;
  - b. тестирование условий;
  - c. тестирование циклов
  - d. тестирование потоков данных
- 3) Выполнение функционального тестирования программ следующими методами:
  - a. разбиение на классы эквивалентности;
  - b. анализ граничных значений;
  - c. анализ причинно-следственных связей;
  - d. предположение об ошибке

### Примеры заданий:

#### Вариант 1

1. Даны цифры двух целых чисел: двузначного  $a_1a_2$  и однозначного  $b$ , где  $a_1$  – число единиц,  $a_2$  – число десятков, Получить цифры числа, равного разности заданных чисел (известно, что это двузначное число).
2. Даны  $N$  целых чисел  $x_1, x_2, \dots, x_N$ . Расставить между ними знак «+» и «-» так, чтобы значение получившегося выражения было равно заданному целому  $S$ .
3. Известно, что любое натуральное число можно представить в виде суммы не более чем четырех квадратов натуральных чисел или, что то же самое, в виде суммы четырех квадратов неотрицательных целых чисел (теорема Лагранжа). Дано натуральное  $n$ ; указать такие неотрицательные целые  $x, y, z, t$ , что  $n = x^2 + y^2 + z^2 + t^2$ .

#### Вариант 2

1. Даны натуральное число  $N$  и вещественные числа  $x_1, x_2, \dots, x_N$ . Выяснить, верно ли, что сумма тех вещественных чисел  $x_i$ , которые меньше 20,5 не превышает 50.
2. Даны  $N$  целых чисел  $x_1, x_2, \dots, x_N$ . Требуется вычеркнуть из них минимальное количество чисел так, чтобы оставшиеся шли в порядке возрастания.
3. Даны натуральные числа  $m, n_1, \dots, n_m$  ( $m > 2$ ). Вычислить НОД( $n_1, \dots, n_m$ ), воспользовавшись для этого соотношением  $\text{НОД}(n_1, \dots, n_k) = \text{НОД}(\text{НОД}(n_1, \dots, n_{k-1}), n_k)$  ( $k = 3, \dots, m$ ) и алгоритмом Евклида

#### Вариант 3

1. Даны натуральное число  $N$  и вещественные числа  $x_1, x_2, \dots, x_N$ . Выяснить, верно ли, что сумма тех вещественных чисел  $x_i$ , которые не превышают 10 кратно 3.
2. Вывести все простые числа от  $M$  до  $N$  включительно.
3. Даны натуральное число  $n$ , целые числа  $a_1, \dots, a_{25}, b_1, \dots, b_n$ . Среди  $a_1, \dots, a_{25}$  нет повторяющихся чисел, нет их и среди  $b_1, \dots, b_n$ . Построить пересечение последовательностей  $a_1, \dots, a_{25}$  и  $b_1, \dots, b_n$  (т. е. получить в каком-нибудь порядке все числа, принадлежащие последовательности  $a_1, \dots, a_{25}$  и последовательности  $b_1, \dots, b_n$  одновременно).

## Тема 6

Работа с классами в C#

### Цель работы:

1. Создать класс. Каждый разрабатываемый класс должен, как правило, содержать следующие элементы:
  - a. скрытые поля;
  - b. конструкторы с параметрами и без параметров,
  - c. методы, свойства. Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса.



- d. при возникновении ошибок должны выбрасываться исключения.
  - e. В программе должна выполняться проверка всех разработанных элементов класса.
2. Создать дочерний класс.
  3. Разработать следующие UML-диаграммы: диаграмма вариантов использования, диаграмма классов, диаграмма деятельности, диаграмма состояния, диаграмма компонентов

### Примеры заданий

#### Вариант 1

Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне.

Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и свойство, позволяющее получить его текущее состояние.

При выходе за границы диапазона выбрасываются исключения. Написать программу, демонстрирующую все разработанные элементы класса.

Создать дочерний класс Счетчик1, который может увеличивать или уменьшать свое значение на любое введенное число в заданном диапазоне.

#### Вариант 2

Описать класс, реализующий шестнадцатеричный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне.

Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и свойство, позволяющее получить его текущее состояние.

При выходе за границы диапазона выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

Создать дочерний класс, который может увеличивать или уменьшать свое значение на любое введенное число в заданном диапазоне

#### Вариант 3

Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол.

Описать свойства для получения состояния объекта.

При невозможности построения треугольника выбрасывается исключение.

Написать программу, демонстрирующую все разработанные элементы класса.

Создать дочерний класс Прямоугольник, который может увеличивать или уменьшать свое значение на любое введенное число в заданном диапазоне

### Тема 7

#### Цель работы:

1. Познакомиться с расширенными возможностями языка программирования C#, такими, как интерфейсы и делегаты.
2. Познакомиться с механизмами событийно-ориентированного программирования на языке C#, такими как механизм обработки событий и исключительные ситуации
3. Разработать объектно-ориентированное, многооконное приложение с использованием интерфейсов, делегатов и событий.
4. Разработать UML - диаграммы
5. Разработать граф диалога пользователя
6. Выполнить оценку качества разработанного приложения по ГОСТ 28195-89

#### Примеры заданий:

Построить иерархию классов в соответствии с вариантом задания. Предусмотреть, по крайней мере, два интерфейса и обработку событий (предусмотреть хотя бы одно событие)

1. Ученик, бакалавр, магистр
2. Студент, преподаватель, заведующий кафедрой
3. Студент, группа, институт
4. Служащий, рабочий, инженер
5. Рабочий, цех, предприятие
6. Рабочий, инженер, администрация
7. Деталь, узел, механизм
8. Организация, завод, страховая компания
9. Журнал, книга, учебник
10. Тест, экзамен, выпускной экзамен
11. Город, область, страна
12. Водоем, озеро, море
13. Растение, куст, дерево
14. Растение, дерево, лесной массив
15. Молочный продукт, сметана, сыр
16. Квитанция, накладная, счет
17. Транспортное средство, автомобиль, поезд
18. Двигатель, двигатель внутреннего сгорания, реактивный двигатель
19. Государство, республика, монархия
20. Животное, млекопитающее, парнокопытное
21. Корабль, парусник, пароход
22. Летательный аппарат, планер, самолет
23. Спортсмен, мастер спорта, заслуженный мастер спорта

## **6 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

### **6.1 Перечень компетенций с указанием этапов их формирования**

При освоении дисциплины формируются следующие компетенции:

- общекультурные ОК-6, ОК-8, ОК-12, ОК-13;
- профессиональные ПК-2, ПК-3, ПК-4, ПК-5, ПК-11.

Указанные компетенции формируются в соответствии со следующими этапами:

1. Формирование и развитие теоретических знаний, предусмотренных указанными компетенциями (лекционные занятия, самостоятельная работа студентов).
2. Приобретение и развитие практических умений, предусмотренных компетенциями (практические занятия, лабораторные работы, выполнение расчетно-графической работы, самостоятельная работа студентов).
3. Закрепление теоретических знаний, умений и практических навыков, предусмотренных компетенциями, в ходе защит лабораторных работ, выполнения расчетно-графической работы, а также решения конкретных технических задач на практических занятиях, успешной сдачи экзамена.

## 6.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описания шкал оценивания

### Образовательные технологии, обеспечивающие результаты освоения дисциплины в форме компетенций

| Код и название компетенции | Компонентный состав компетенции (дескрипторы)  | Технологии формирования      | Средства оценки           |
|----------------------------|--|------------------------------|---------------------------|
| ОК-6                       | Знать: Основы языка программирования C#  | лекции, лабораторные занятия | Опрос, контрольные работы |
|                            | Уметь: Уметь разрабатывать приложения на языке программирования C#   | лабораторные занятия         | Опрос, контрольные работы |
|                            | Владеть: навыками работы с массивами, коллекциями, строками, файлам и файловыми потоками на языке программирования C# при разработке конкретных прикладных задач   | лабораторные занятия         | Опрос, контрольные работы |
| ОК-8                       | Знать: методiku оценки качества разработанного программного обеспечения по ГОСТ 28195.89   | лекции, лабораторные занятия | Опрос                     |
|                            | Уметь: подбирать оценочные элементы при оценке качества по ГОСТ 28195.89   | лабораторные занятия         | Опрос                     |
|                            | Владеть: практическими навыками оценки качества разработанного программного обеспечения по ГОСТ 28195.89, сравнивать полученные данные с базовыми образцами        | лабораторные занятия         | Опрос                     |
| ОК-12                      | Знать: Основы работы в среде Visual Studio   | лекции, лабораторные занятия | Опрос, контрольные работы |
|                            | Уметь: разрабатывать приложения в среде Visual Studio  | лабораторные занятия         | Опрос                     |
|                            | Владеть: навыками тестирования и отладки разработанных приложений в среде Visual Studio  | лабораторные занятия         | Опрос                     |
| ОК-13                      | Знать: Принципы построения современных многомодульных, многооконных приложений, методы межпроцессного взаимодействия и передачи сообщений между приложениями.      | лекции, лабораторные занятия | Опрос, контрольные работы |
|                            | Уметь: создавать многооконные, многомодульные приложения, организовывать передачу сообщений между приложениями   | лабораторные занятия         | Опрос                     |
|                            | Владеть: методами разработки многооконных, многомодульных приложений, методами создания программных интерфейсов, разработкой делегатов и программированием событий | лабораторные занятия         | Опрос                     |
| ПК-2                       | Знать: Методы структурного, объектно-ориентированного программирования и   | лекции, лабораторные         | Опрос                     |

|       |   |                              |                            |
|-------|---|------------------------------|----------------------------|
|       | проектирования. Методы тестирования и отладки приложений.   | занятия                      |                            |
|       | Уметь: Проектировать функциональные диаграммы, диаграммы потоков данных, диаграммы переходов состояний, структурные и функциональные схемы. Уметь создавать UML-диаграммы. Уметь использовать CASE-средства для структурного и объектно-ориентированного проектирования | лабораторные занятия         | Опрос                      |
|       | Владеть: методами использования методологии SADT и языка UML для создания программных проектов  | лабораторные занятия         | Опрос                      |
| ПК-3  | Знать: Принципы построения современных многомодульных, многооконных приложений, методы межпроцессного взаимодействия и передачи сообщений между приложениями.   | лекции, лабораторные занятия | Опрос                      |
|       | Уметь: создавать многооконные, многомодульные приложения, организовывать передачу сообщений между приложениями  | лабораторные занятия         | Опрос                      |
|       | Владеть: методами разработки многооконных, многомодульных приложений, методами создания программных интерфейсов, разработкой делегатов и программированием событий  | лабораторные занятия         | Опрос                      |
| ПК-4  | Знать: Основы объектно-ориентированного программирования и языка UML  | лекции, лабораторные занятия | Опрос                      |
|       | Уметь: Создавать UML-диаграммы  | лабораторные занятия         | Опрос                      |
|       | Владеть: Методами использования UML-языка для разработки практических задач   | лабораторные занятия         | Опрос                      |
| ПК-5  | Знать: Основы работы с CASE средствами для структурного и объектно-ориентированного проектирования (BPWin, Microsoft Office Visio). Знать основы работы с CASE-средствами для создания баз данных (ERWin). Знать основные пункты технического задания                   | лекции, лабораторные занятия | Опрос, контрольные работы, |
|       | Уметь: Уметь проводить предпроектные исследования и составлять техническое задание. Работать в среде ERWin, BPWin, Microsoft Office Visio   | лабораторные занятия         | Опрос, контрольные работы  |
|       | Владеть: CASE-средствами при проектировании структурных и объектно-ориентированных приложений, а также CASE-средствами для работы с базами данных.  | лабораторные занятия         | Опрос, контрольные работы  |
| ПК-11 | Знать: требования к разработке технологичных продуктов, типы сцепления и связности модулей. Требования к оформлению программных продуктов   | лекции, лабораторные занятия | Опрос, контрольная работа  |
|       | Уметь: Определять тип сцепления и связность   | лабораторные                 | Опрос,                     |

|  |   |                      |                           |
|--|---|----------------------|---------------------------|
|  | модулей, разрабатывать приложения, удовлетворяющие требованиям эффективности по времени выполнения и требуемой памяти | занятия              | контрольная работа        |
|  | Владеть: Методикой оценки степени технологичности программных продуктов   | лабораторные занятия | Опрос, контрольная работа |

**Оценка уровней сформированности компетенций в результате освоения учебной дисциплины**

| Коды компетенций<br>Название компетенции | Уровни сформированности компетенции      | Основные признаки уровня   |
|--|--|--|
| <b>Общекультурные компетенции - ОК</b>   |  |  |
| ОК-6                                     | Пороговый уровень освоения компетенции   | Знает: Основы языка программирования С#  |
|  | Продвинутый уровень освоения компетенции | Дополнительно умеет: Разрабатывать приложения на языке программирования С#   |
|  | Высокий уровень освоения компетенции     | Дополнительно владеет: навыками работы с массивами, коллекциями, строками, файлам и файловыми потоками на языке программирования С# при разработке конкретных прикладных задач |
| ОК-8                                     | Пороговый уровень освоения компетенции   | Знает: методику оценки качества разработанного программного обеспечения по ГОСТ 28195.89   |
|  | Продвинутый уровень освоения компетенции | Дополнительно умеет: подбирать оценочные элементы при оценке качества по ГОСТ 28195.89   |
|  | Высокий уровень освоения компетенции     | Дополнительно владеет: практическими навыками оценки качества разработанного программного обеспечения по ГОСТ 28195.89, сравнивать полученные данные с базовыми образцами      |
| ОК-12                                    | Пороговый уровень освоения компетенции   | Знает: Основы работы в среде Visual Studio   |
|  | Продвинутый уровень освоения компетенции | Дополнительно умеет: разрабатывать приложения в среде Visual Studio  |
|  | Высокий уровень освоения компетенции     | Дополнительно владеет: навыками тестирования и отладки разработанных приложений в среде Visual Studio  |
| ОК-13                                    | Пороговый уровень освоения компетенции   | Знает: Принципы построения современных многомодульных, многооконных приложений, методы межпроцессного взаимодействия и передачи сообщений между приложениями                   |
|  | Продвинутый уровень освоения компетенции | Дополнительно умеет: создавать многооконные, многомодульные приложения, организовывать передачу сообщений между приложениями   |
|  | Высокий уровень освоения компетенции     | Дополнительно владеет: навыками разработки многооконных, многомодульных приложений, методами создания программных интерфейсов,   |

|  |  |   |
|--|--|---|
|  |  | разработкой делегатов и программированием событий   |
| <b>Профессиональные компетенции - ПК</b> |  |   |
| ПК-2                                     | Пороговый уровень освоения компетенции   | Знает: Методы структурного, объектно-ориентированного программирования и проектирования. Методы тестирования и отладки приложений.  |
|  | Продвинутый уровень освоения компетенции | Дополнительно умеет: Проектировать функциональные диаграммы, диаграммы потоков данных, диаграммы переходов состояний, структурные и функциональные схемы. Умеет создавать UML-диаграммы. Умеет использовать CASE-средства для структурного и объектно-ориентированного проектирования |
|  | Высокий уровень освоения компетенции     | Дополнительно владеет: навыками использования методологии SADT и языка UML для создания программных проектов  |
| ПК-3                                     | Пороговый уровень освоения компетенции   | Знает: Принципы построения современных многомодульных, многооконных приложений, методы межпроцессного взаимодействия и передачи сообщений между приложениями  |
|  | Продвинутый уровень освоения компетенции | Дополнительно умеет: создавать многооконные, многомодульные приложения, организовывать передачу сообщений между приложениями  |
|  | Высокий уровень освоения компетенции     | Дополнительно владеет: навыками разработки многооконных, многомодульных приложений, методами создания программных интерфейсов, навыками разработки делегатов и программирования событий   |
| ПК-4                                     | Пороговый уровень освоения компетенции   | Знает: Основы объектно-ориентированного программирования и языка UML  |
|  | Продвинутый уровень освоения компетенции | Дополнительно умеет: Создавать UML-диаграммы  |
|  | Высокий уровень освоения компетенции     | Дополнительно владеет: навыками использования UML-языка для разработки практических задач   |
| ПК-5                                     | Пороговый уровень освоения компетенции   | Знает: Основы работы с CASE средствами для структурного и объектно-ориентированного проектирования (BPWin, Microsoft Office Visio). Знает основы работы с CASE-средствами для создания баз данных (ERWin). Знает основные пункты технического задания                                 |
|  | Продвинутый уровень освоения компетенции | Дополнительно умеет: проводить предпроектные исследования и составлять техническое задание. Работать в среде ERWin, BPWin, Microsoft Office Visio   |
|  | Высокий уровень освоения компетенции     | Дополнительно владеет: навыками использования CASE- средств при проектировании структурных и объектно-ориентированных приложений, а также CASE- средств для работы с базами   |



|       |  |   |
|-------|--|---|
|       |  | данных.   |
| ПК-11 | Пороговый уровень освоения компетенции   | Знает: требования к разработке технологичных продуктов, типы сцепления и связности модулей. Требования к оформлению программных продуктов                                       |
|       | Продвинутый уровень освоения компетенции | Дополнительно умеет: Определять тип сцепления и связность модулей, разрабатывать приложения, удовлетворяющие требованиям эффективности по времени выполнения и требуемой памяти |
|       | Высокий уровень освоения компетенции     | Дополнительно владеет: навыками оценки степени технологичности программных продуктов  |

### Формы текущего контроля по разделам, темам дисциплины

| №№пп | Наименование раздела, темы дисциплины | Формы текущего контроля |
|------|---------------------------------------|-------------------------|
| 1.   | Тема 1                                | Тест                    |
| 2.   | Тема 2                                | Контрольная работа      |
| 3.   | Тема 3                                | Контрольная работа      |
| 4.   | Тема 4                                | Опрос                   |
| 5.   | Тема 5                                | Контрольная работа      |
| 6.   | Тема 6                                | Контрольная работа      |
| 7.   | Тема 7                                | Опрос                   |
| 8.   | Тема 8                                | Опрос                   |

### 6.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

#### Вопросы к экзамену

1. Технология программирования. Основные понятия. Этапы развития технологии программирования.
2. Жизненный цикл и этапы разработки программного обеспечения
3. Модели жизненного цикла программного обеспечения
4. Понятия эффективности и технологичности программного обеспечения. Модули и их свойства (сцепление и связность модулей)
5. Структурное программирование
6. Средства описания структурных алгоритмов (псевдокоды, схемы алгоритмов)
7. Средства описания структурных алгоритмов (Flow-формы, диаграммы Насси-Шнейдермана)
8. Правила оформления программ
9. Разработка технического задания
10. Классификация моделей разрабатываемого программного обеспечения
11. Структурный подход. Диаграммы переходов состояний
12. Структурный подход. Функциональные диаграммы
13. Структурный подход. Диаграммы потоков данных
14. Структурный подход. Структуры данных и диаграммы отношений компонентов данных
15. Структурный подход. Сетевая модель данных (Диаграммы «сущность-связь»)

16. Проектирование программного обеспечения при структурном подходе. Структурная и функциональная схемы
17. Структурный подход. Структурные карты Константайна
18. Проектирование структур данных. Методика Джексона
19. UML- стандартный язык описания разработки программных продуктов с использованием объектного подхода
20. Диаграммы вариантов использования
21. Диаграмма классов. Отношения между классами
22. Диаграмма последовательностей
23. Диаграммы деятельности
24. Диаграмма пакетов
25. Диаграммы состояний объекта
26. Диаграмма кооперации
27. Диаграмма компонентов
28. Диаграмма размещения
29. Структурное тестирование. Тестирование базового пути
30. Структурное тестирование. Тестирование условий
31. Структурное тестирование. Тестирование циклов
32. Структурное тестирование. Тестирование потоков данных
33. Функциональное тестирование. Разбиение на классы эквивалентности и анализ граничных значений
34. Функциональное тестирование. Анализ причинно-следственных связей
35. Классификация ошибок
36. Методы отладки программного обеспечения
37. Разработка пользовательского интерфейса. Классификация диалогов и общие принципы их работы
38. Разработка пользовательского интерфейса. Граф диалога с пользователем
39. Оценка качества программного обеспечения по ГОСТ 28195-89

### **Примеры экзаменационных задач**

#### **ВАРИАНТ 1**

Класс Деньги для работы с денежными суммами. Число должно быть представлено двумя полями: для рублей и для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение и вычитание.

Создать дочерний класс, который выполняет деление, умножение и операцию сравнения.

Проверить работу созданных методов.

#### **ВАРИАНТ 2**

Класс Равнобокая трапеция, члены класса: координаты 4-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура равнобокой трапецией.

Дочерний класс выполняет вычисления и вывод сведений о фигуре: длины сторон, периметр, площадь.

Продемонстрировать работу с классом: дано N трапеций, найти количество трапеций, у которых площадь больше средней площади.

#### **ВАРИАНТ 3**

С помощью регулярных выражений проверить, надежно ли составлен пароль. Пароль считается надежным, если он состоит из 8 или более символов. Где символом может быть английская буква, цифра и знак подчеркивания. Пароль должен содержать хотя бы одну заглавную букву, одну маленькую букву и одну цифру.

Пример правильных выражений:



C001\_Pass  
SupperPas1

Пример неправильных выражений:

Cool\_pass  
C001

#### ВАРИАНТ 4

Есть строка вида 159,5|15|19.6

Требования:

- Количество чисел может быть любым
- Дробная часть числа может отделяться как точкой, так и запятой.
- Строка не должна содержать в начале и в конце символ "|" (т.е. должна начинаться и заканчиваться цифрами)
- Строка не должна содержать никаких лишних символов

С помощью регулярных выражений для любой введенной строки проверить соответствие требованиям.

#### Примеры тестов для промежуточной аттестации

##### Тема 1. Основы программирования в среде C#

1. Для чего служат строки, начинающиеся с двух или трех косых черт:

С таких строк начинается описание класса; Служат для обозначения ключевых слов; Такие строки являются комментариями и служат для документирования текста программы; Такие строки ограничивают блок операторов, выполняющихся в циклах или в операторах ветвления

2. Чему будет равно значение переменных a, b и c после выполнения следующего фрагмента программы?

```
{ int a = 0; int b = 0; int c = 0;
  a = (b = 2 + 3) / 2 - 4 + (c = 5 % 2);
  Console.WriteLine("a={0} b={1} c={2}", a, b, c);}
```

3. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
{   const int LIMIT = 10;
    int i = 0;
    while (i < LIMIT)
    Console.WriteLine("Привет!");
    i++; }
```

4. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
{   const int LIMIT = 10;
    int i = 0;
    while (i < LIMIT){
    Console.WriteLine("Привет!");
    i++;}}
```

5. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
{   const int LIMIT = 10;
    uint i = 0;
    while (i < LIMIT)
    {   Console.WriteLine("Привет!");
        i=i-1;}}
```

6. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
{
    const int LIMIT = 10;
    for (uint i = 0; i <= LIMIT; ++i)
        Console.WriteLine("Привет!");
}
```

7. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
{
    const int LIMIT = 10;
    uint i = 1;
    do
    {
        Console.WriteLine("Привет!");
        i=i+1;
    } while (i>LIMIT);
}
```

8. Чему будет равна переменная i после завершения работы цикла?

```
static void Main(string[] args)
{
    const int LIMIT = 10;
    uint i = 0;
    for (i = 0; i <= LIMIT; ++i)
        Console.WriteLine("Привет!");
    Console.WriteLine("i={0}", i);
}
```

9. Чему будет равна переменная a после завершения работы программы?

```
static void Main(string[] args)
{
    bool i = false;
    int a = 0;
    if (i = true) a = 3;
    else a = -3;
    Console.WriteLine("a={0}", a);
}
```

10. Чему будет равна переменная i после завершения работы программы?

```
static void Main(string[] args)
{
    int i = -1;
    if (i < 0 && (i!==-1)) i = 3;
    else i = -3;
    Console.WriteLine("i={0}", i);
}
```

11. Чему будет равна переменная i после завершения работы программы?

```
static void Main(string[] args)
{
    int i = 0;
    if ((i==i) || (i==2)) i = 3;
    else i = -3;
    Console.WriteLine("i={0}", i);
}
```

12. Чему будет равна переменная i после завершения работы программы?

```
static void Main(string[] args)
{
    int a=8; int b=12; int i =0;
    i=((2*b-2*8)+2)/3;
    if (i-3>0)
        if (i==3) i=1;
        else i = 2;
    Console.WriteLine("i={0}", i);
}
```

13. Чему будет равно значение переменной a после завершения работы программы?

```
static void Main(string[] args)
{
    int a=8; int b=12; bool i =true;
    if (i == a > b) a = 1;
}
```

```
else
    if (i == !(a != b)) a = 2;
    else if (i == (a < b)) a = 3;
    else a = 4;
    Console.WriteLine("a={0}", a);}
```

14. Чему будет равно значение переменной *a* после завершения работы программы?

```
static void Main(string[] args)
{
    int a=-10;
    a = (a + 1 > 0) ? a : -a;
    Console.WriteLine("a={0}", a);}
```

15. Чему будет равно значение переменной *b* после завершения работы программы?

```
static void Main(string[] args)
{
    int b=-9;
    b = (b + 1 >= 0) ? b : -b;
    Console.WriteLine("b={0}", b);}
```

16. Чему будет равно значение переменной *c* после завершения работы программы?

```
static void Main(string[] args)
{
    int c = -10; int a = 0;
    a = (c + 1 >= 0) ? c : -c;
    Console.WriteLine("c={0}", c);}
```

17. Чему будет равно значение переменной *d* после завершения работы программы?

```
static void Main(string[] args)
{
    int d = -10;
    d+= (d + 1 > 0) ? d : -d;
    Console.WriteLine("d={0}", d);}
```

18. Чему будет равно значение переменной *e* после завершения работы программы?

```
static void Main(string[] args)
{
    int e = -10;
    e = (--e > 0) ? e : -e;
    Console.WriteLine("e={0}", e);}
```

19. Чему будет равно значение переменной *s* после завершения работы программы?

```
static void Main(string[] args)
{
    int i = 5;
    float s = 1;
    s*=(i+=5)/4;
    Console.WriteLine("s={0}", s);}
```

20. Чему будет равно значение переменной *s* после завершения работы программы?

```
static void Main(string[] args)
{
    int i = 5;
    float s = 1;
    s*=(float)(i+=5)/4;
    Console.WriteLine("s={0}", s);}
```

21. Чему будет равно значение переменной *i* после завершения работы программы?

```
static void Main(string[] args)
{
    int i = 0; int s = 10;
    for (i = 0; s > 0; i++)
        s /= 2;
    Console.WriteLine("i={0}", i);}
```

22. Чему будет равно значение переменной *j* после завершения работы программы?

```
static void Main(string[] args)
{
    int j = 0; int s = 10;
```

```
for (j = 0; s > 0 && (s /= 2) != 0; j++);  
Console.WriteLine("j={0}", j);}
```

23. Чему будет равно значение переменной j после завершения работы программы?

```
static void Main(string[] args)  
{ int j = 0; int s = 10;  
for (j = 0; s > 0 && (s %= 2) == 0; j++);  
Console.WriteLine("j={0}", j);}
```

24. Чему будет равно значение переменной k (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)  
{ int i = 0x1234; int k = 0;  
k=i<<4;  
Console.WriteLine("k={0:x}", k);}
```

25. Чему будет равно значение переменной j (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)  
{ int i = 0x1234; int j = 0;  
j=i>>8;  
Console.WriteLine("j={0:x}", j); }
```

26. Чему будет равно значение переменной r (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)  
{ int i = 0x45ff; int j = 0x00ff;  
int r =0;  
r=i^j;  
Console.WriteLine("r={0:x}", r);}
```

27. Чему будет равно значение переменной r (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)  
{ int i = 0x45ff; int j = 0x00ff;  
int r =0;  
r=i|j;  
Console.WriteLine("r={0:x}", r);}
```

28. Чему будет равно значение переменной r (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)  
{ int i = 0x45ff; int j = 0x00ff;  
int r =0;  
r=i&j;  
Console.WriteLine("r={0:x}", r);}
```

29. Чему будет равно значение переменной a после завершения работы программы?

```
static void Main(string[] args)  
{ int a = 5; int b = 2;  
switch (b++)  
{ case 1: a += 3; break;  
case 2: a *= 3; break;  
case 3: a = a * 2 + 10; break;  
case 4: a %= 6; break; }  
Console.WriteLine("a={0}", a);}
```

30. Чему будет равно значение переменных a и b после завершения работы программы?

```
static void Main(string[] args)
```

```
{ int a = 5; int b = 2; int c = 30; int d = 7;  
  a += ++b+c;  
  c -= d++ + ++b;  
  Console.WriteLine("a={0}      c={1}", a,c);}
```

## Примеры контрольных работ

### Контрольная работа по Теме 2

**Основные понятия и подходы. Приемы обеспечения технологичности программных продуктов**

#### Вопросы:

1. Дайте определение технологичности программного обеспечения
2. Что понимают под связностью модуля, типы связности
3. Что понимают под сцеплением модуля, типы сцепления
4. Нисходящая и восходящая разработка программного обеспечения. Перечислите достоинства и недостатки данных методов.
5. Постройте диаграмму Насси-Шнейдермана для программы перевода числа из десятичной формы в шестнадцатеричную
6. Перечислите формы описания структурных алгоритмов, опишите особенности данных форм, их достоинства и недостатки
7. Постройте flow-форму для программы определения типа треугольника по трем его сторонам
8. Опишите метод оценки характеристик разработанных программ с помощью метрик Холстеда
9. Опишите метод оценки характеристик разработанных программ с помощью метрик Джилба
10. Опишите метод оценки надежности программных средств с помощью модели Джелиински – Моранды
11. Опишите метод оценки надежности программных средств с помощью модели Нельсона
12. Опишите метод оценки надежности программных средств с помощью эвристической модели

### Контрольная работа по Теме 3.

#### Разработка технического задания

1. Назовите основные эксплуатационные требования к программным продуктам. Какими средствами и приемами обеспечивается каждый из них? Для каких типов программных систем целесообразно указывать каждый из них?
2. В каких ситуациях необходимы предпроектные исследования? Какие вопросы при этом решают? Что получают в результате таких исследований?
3. Назовите, какой раздел технического задания можно считать основным и почему? Какую информацию должны содержать остальные разделы? В чем основная сложность разработки технического задания?
4. Составьте техническое задание на разработку «калькулятора» по типу, предлагаемого Windows.
5. Разработать программный модуль «Учет успеваемости стуков». Программный модуль предназначен для оперативного успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентоов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.
6. Разработать программный модуль «Личные дела студентов. Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.
7. Разработать программный модуль «Решение комбинаторно-оптимизационных задач». Модуль должен содержать алгоритмы поиска цикла минимальной длины (задача коммивояжера), поиска кратчайшего пути и поиска минимального связывающего дерева.
8. Разработать приложение Windows «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.
9. Разработать приложение Windows «Калькулятор». Приложение предназначено для любых пользователей и должно содержать все арифметические операции (с соблюдением приоритетов) и желательно (но не обязательно) несколько математических функций.
10. Разработать программный модуль «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.). Модуль предназначен для использования сотрудниками отдела кадров и деканата.

11. Разработать программный модуль «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень). Модуль предназначен для использования сотрудниками профкома и отдела кадров.
12. Разработать программный модуль «Автосервис». При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, марка автомобиля, вид работы, дата приема заказа и стоимость ремонта. После выполнения работ распечатывается квитанция.
13. Разработать программный модуль «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.
14. Разработать программный модуль «Картотека агентства недвижимости», предназначенный для использования работниками агентства. В базе содержатся сведения о квартирах (количество комнат, этаж, метраж и др.). При поступлении заявки на обмен (куплю, продажу) производится поиск подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.
15. Разработать программный модуль «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.
16. Разработать программный модуль «Авиакасса», содержащий сведения о наличии свободных мест на авиамаршруты. В базе должны содержаться сведения о номере рейса, экипаже, типе самолета, дате и времени вылета, а также стоимости авиабилетов (разного класса). При поступлении заявки на билеты программа производит поиск подходящего рейса.
17. Разработать программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.
18. Разработать программный модуль «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.
19. Разработать программный модуль «Кадровое агентство», содержащий сведения о вакансиях и резюме. Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

### **Контрольная работа по Теме 5. Тестирование и отладка программных продуктов при структурном подходе**

1. Определите понятие тестирования.
2. Что такое тест? Поясните содержание процесса тестирования.
3. Что такое исчерпывающее тестирование?
4. Какие задачи решает тестирование?
5. Каких задач не решает тестирование?
6. Какие принципы тестирования вы знаете? В чем их отличие друг от друга?
7. В чем состоит суть тестирования «черного ящика»?
8. В чем состоит суть тестирования «белого ящика»?
9. Каковы особенности тестирования «белого ящика»?
10. Какие недостатки имеет тестирование «белого ящика»?
11. Какие достоинства имеет тестирование «белого ящика»?
12. Дайте характеристику способа тестирования базового пути.
13. Какие особенности имеет потоковый граф?
14. Поясните понятие независимого пути.
15. Поясните понятие цикломатической сложности.
16. Что такое базовое множество?
17. Какие свойства имеет базовое множество?
18. Какие способы вычисления цикломатической сложности вы знаете?
19. Поясните шаги способа тестирования базового пути.
20. Поясните достоинства, недостатки и область применения способа тестирования базового пути.
21. Дайте общую характеристику способов тестирования условий.

22. Какие типы ошибок в условиях вы знаете?
23. Какие методики тестирования условий вы знаете?
24. Поясните суть способа тестирования ветвей и операторов отношений. Какие он имеет
25. ограничения?
26. Каковы особенности тестирования методом «черного ящика»?
27. Какие категории ошибок выявляет тестирование методом «черного ящика»?
28. Какие достоинства имеет тестирование методом «черного ящика»?
29. Поясните суть способа разбиения по эквивалентности.
30. Что такое класс эквивалентности?
31. Какие правила формирования классов эквивалентности вы знаете?
32. Как выбирается тестовый вариант при тестировании по способу разбиения по эквивалентности?
33. Поясните суть способа анализа граничных значений.
34. Чем способ анализа граничных значений отличается от разбиения по эквивалентности?
35. Поясните правила анализа граничных значений.
36. то такое дерево разбиений? Каковы его особенности?
37. В чем суть способа диаграмм причин-следствий?
38. Что такое причина?
39. Что такое следствие?
40. Дайте общую характеристику графа причинно-следственных связей.
41. Какие функции используются в графе причин и следствий?
42. Какие ограничения используются в графе причин и следствий?
43. Поясните шаги способа диаграмм причин-следствий.
44. Какую структуру имеет таблица решений в способе диаграмм причин-следствий?
45. Как таблица решений преобразуется в тестовые варианты?

### **Контрольная работа по Теме 6**

#### **Проектирование программного обеспечения при объектно-ориентированном подходе к программированию**

1. Какие элементы определяются в составе класса?
2. Приведите синтаксис описания класса в общем виде. Проиллюстрируйте его фрагментом программы на языке C#.
3. Какие модификаторы типа доступа Вам известны?
4. В чем заключаются особенности доступа членов класса с модификатором public, private, protected, internal?
5. Приведите синтаксис создания объекта в общем виде. Проиллюстрируйте его фрагментом программы на языке C#.
6. Что понимается под термином «конструктор», В чем состоит назначение конструктора, Каждый ли класс языка C# имеет конструктор, Какие умолчания для конструкторов приняты в языке C#? Приведите синтаксис конструктора класса в общем виде. Проиллюстрируйте его фрагментом программы на языке C#.
7. Что понимается под термином «деструктор», В чем состоит назначение деструктора? Приведите синтаксис деструктора класса в общем виде. Проиллюстрируйте его фрагментом программы на языке C#.
8. Что понимается под термином «наследование»? Какая классификация объектов соответствует наследованию? Что общего имеет дочерний класс с родительским? В чем состоит различие между дочерним и родительским классами? Приведите синтаксис описания наследования классов в общем виде. Проиллюстрируйте его фрагментом программы на языке C#.



#### **6.4 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Процедуры оценивания знаний, умений, навыков, характеризующих этапы формирования компетенций, изложены в:

1. Конспект лекций по дисциплине (см. приложение 3.РПД Б3.В.ОД.6 (лк));
2. Методические указания к выполнению лабораторных работ (см. приложение 3.РПД Б3.В.ОД.6 (лб));
3. Методические указания к самостоятельной работе студентов (см. приложение 3.РПД Б3.В.ОД.6 (ср)).

#### **7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

##### **Основная учебная литература**

1. Буч Г. Язык UML. Руководство пользователя [Электронный ресурс] : / Буч Г., Рамбо Д., Якобсон И. — Электрон. дан. — М. : ДМК Пресс, 2008. — 494 с. — Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=1246](http://e.lanbook.com/books/element.php?pl1_id=1246)
2. Вишневская, Т.И. Технология программирования. Часть 1 [Электронный ресурс] : учебно-методическое пособие / Т.И. Вишневская, Т.Н. Романова. — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана (Московский государственный технический университет имени Н.Э. Баумана), 2007. — 59 с. — Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=52381](http://e.lanbook.com/books/element.php?pl1_id=52381)
3. Вишневская, Т.И. Технология программирования. Часть 2 [Электронный ресурс] : учебно-методическое пособие / Т.И. Вишневская, Т.Н. Романова. — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана (Московский государственный технический университет имени Н.Э. Баумана), 2010. — 52 с. — Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=52411](http://e.lanbook.com/books/element.php?pl1_id=52411)
4. Гаврилов А.И., Малахов В.В. Малащенко И.В., Панкратова Е.А., Семенова О.В. Основы программирования в среде C#. Методические указания к лабораторным работам по дисциплине «Технологии программирования» – Смоленск: РИО филиала МЭИ в г.Смоленске, 2014. – 110 с.
5. Малахов В.В., Панкратова Е.А. Семенова О.В. Проектирование программного обеспечения. – Смоленск: РИО филиала МЭИ в г.Смоленске, 2010. – 2,25 п.л.
6. Панкратова Е.А., Семенова О.В. Тестирование программного обеспечения. – Смоленск: РИО филиала МЭИ в г.Смоленске, 2011. – 125 п.л.

##### **Дополнительная учебная литература**

7. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. Учебник для студентов ВУЗов 2-е издание, переработанное и дополненное –М.: «Финансы и статистика», 2006 г. -534 с.:ил
8. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем. Учебник для студентов ВУЗов. –М.: «Финансы и статистика», 2006 г. -190 с. :ил.
9. Вилле К. Представляем C# [Электронный ресурс] : . — Электрон. дан. — М. : ДМК Пресс, 2008. — 186 с. — Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=1225](http://e.lanbook.com/books/element.php?pl1_id=1225)
10. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие / под ред. Л.Г. Гагариной –М: ИД «ФОРУМ»: ИНФРА-М, 2008. –400с.: ил. – (Высшее образование)
11. Иванова Г.С. Технология программирования: учебник. –М. КНОРУС, 2011 336 с.
12. Орлов С.А. Технологии разработки программного обеспечения: Учебник. –СПб. Питер, 2012, 464 с.: ил.
13. Робисон У. C# без лишних слов [Электронный ресурс] : . — Электрон. дан. — М. : ДМК



Пресс, 2008. — 342 с. — Режим доступа:  
[http://e.lanbook.com/books/element.php?pl1\\_id=1240](http://e.lanbook.com/books/element.php?pl1_id=1240)

14. Черников Б.В., Поклонов Б.Е. Оценка качества программного обеспечения: Практикум; учебное пособие. –М: ИД «ФОРУМ»: ИНФРА-М, 2012, -400 с.: ил.

## **8 Перечень ресурсов информационно-телекоммуникационной сети «Интернет» необходимых для освоения дисциплины**

1. <http://www.rugost.com/index.php> - ГОСТ 19.xxx Единая система программной документации (ЕСПД)
2. [http://standartgost.ru/0/2870/2880-edinaya\\_sistema\\_programmnoy\\_dokumentatsii](http://standartgost.ru/0/2870/2880-edinaya_sistema_programmnoy_dokumentatsii) - ГОСТ 19.xxx Единая система программной документации (ЕСПД)

## **9 Методические указания для обучающихся по освоению дисциплины**

Дисциплина предусматривает лекции раз в неделю, и лабораторные работы раз в две недели. Изучение курса завершается экзаменом.

Успешное изучение курса требует посещения лекций, активной работы на лабораторных работах, выполнения всех учебных заданий преподавателя, ознакомления с основной и дополнительной литературой.

Во время **лекции** студент должен вести краткий конспект.

Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или на ближайшей лекции.

Обучающемуся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

**Практические (семинарские) занятия** составляют важную часть профессиональной подготовки студентов. Основная цель проведения практических (семинарских) занятий - формирование у студентов аналитического, творческого мышления путем приобретения практических навыков.

Методические указания к практическим (семинарским) занятиям по дисциплине наряду с рабочей программой и графиком учебного процесса относятся к методическим документам, определяющим уровень организации и качества образовательного процесса.

Содержание *практических (семинарских) занятий* фиксируется в РПД в разделе 4 настоящей программы.

Важнейшей составляющей любой формы практических занятий являются упражнения (задания). Основа в упражнении - пример, который разбирается с позиций теории, развитой в лекции. Как правило, основное внимание уделяется формированию конкретных умений, навыков, что и определяет содержание деятельности студентов - решение задач, графические работы, уточнение категорий и понятий науки, являющихся предпосылкой правильного мышления и речи.

**Лабораторные работы** составляют важную часть профессиональной подготовки студентов. Они направлены на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений.

Выполнение студентами лабораторных работ направлено на:

обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам дисциплин;

формирование необходимых профессиональных умений и навыков;

Содержание лабораторных работ фиксируется в РПД в разделе 4 настоящей программы.

При планировании лабораторных работ следует учитывать, что наряду с ведущей целью - подтверждением теоретических положений - в ходе выполнения заданий у студентов формируются практические умения и навыки обращения с лабораторным оборудованием, аппаратурой и пр., которые могут составлять часть профессиональной практической подготовки, а также исследовательские умения (наблюдать, сравнивать, анализировать, устанавливать зависимости, делать выводы и обобщения, самостоятельно вести исследование, оформлять результаты).

Состав заданий для лабораторной работы +спланирован с таким расчетом, чтобы за отведенное время они могли быть качественно выполнены большинством студентов.

Выполнению лабораторных работ предшествует проверка знаний студентов – их теоретической готовности к выполнению задания.

Помимо собственно выполнения работы для каждой лабораторной работы предусмотрена процедура защиты, в ходе которой преподаватель проводит устный или письменный опрос студентов для контроля понимания выполненных ими измерений, правильной интерпретации полученных результатов и усвоения ими основных теоретических и практических знаний по теме занятия.

При подготовке к экзамену в дополнение к изучению конспектов лекций и учебных пособий, необходимо пользоваться учебной литературой, рекомендованной к настоящей программе. При подготовке к экзамену нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по несколько типовых задач из каждой темы. При решении задач всегда необходимо уметь качественно интерпретировать итог решения.

**Самостоятельная работа студентов (СРС)** по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации для обеспечения СРС готовятся преподавателем и являются неотъемлемой частью программы.

#### **10 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

При проведении лабораторных работ предусматривается использование персональных компьютеров, оснащенных необходимым комплектом лицензионного программного обеспечения. – Visual Studio 2010 по подписке Dream Spark и Microsoft Visio по подписке Dream Spark.

#### **11 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

##### **Лекционные занятия:**

Аудитория.

**Лабораторные работы** по данной дисциплине проводятся в компьютерных классах, оснащенных необходимым комплектом программного обеспечения.

Автор  
канд. техн. наук, доцент

Е.А. Панкратова

Зав. кафедрой ВТ  
д-р техн. наук, профессор

А.С. Федулов

Программа одобрена на заседании кафедры 28 августа 2015 года, протокол № 01.