

Направление подготовки 09.03.01 «Информатика и вычислительная техника»
Профиль «Вычислительные машины, комплексы, системы и сети»
РПД Б1.В.03 «Технология программирования»



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

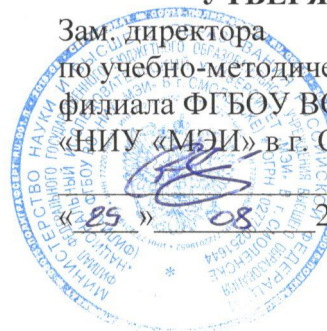
РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

(НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ)

УТВЕРЖДАЮ

Зам. директора
по учебно-методической работе
филиала ФГБОУ ВО
«НИУ «МЭИ» в г. Смоленске
В.В. Рожков
« 25 » 08 20 18 г.



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

Профиль **«Вычислительные машины, комплексы, системы и сети»**

Уровень высшего образования: **бакалавриат**

Нормативный срок обучения: **4 года 11 месяцев**

Форма обучения: **заочная**

Год набора: **2018**

Смоленск – 2018 г.

Программа составлена с учетом ФГОС ВО – бакалавриата по направлению подготовки 09.03.01 «Информатика и вычислительная техника», утвержденного приказом Минобрнауки России от «19» сентября 2017 г. № 929

Программу составил:
к.т.н., доцент кафедры «Вычислительная техника»

Я.А. Федулов

«25» июня 2018 г.

Программа обсуждена и одобрена на заседании кафедры «Вычислительная техника»
«26» июня 2018 г., протокол № 10

Заведующий кафедрой вычислительной техники
д.т.н., профессор

А.С. Федулов

«02» июля 2018 г.

РПД адаптирована для лиц с ограниченными возможностями здоровья и инвалидов

Ответственный в филиале по работе
с ЛОВЗ и инвалидами

Е.В. Зуева

«02» июля 2018 г.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины «Технология программирования» является формирование знаний, умений и навыков в определении требований к информационным системам и программному обеспечению и проектировании его на основе современных технологий программирования.

Задачами дисциплины являются: изучение основных понятий и определений, классификаций программного обеспечения; получение знаний об основных этапах создания программного обеспечения в рамках жизненного цикла, при современном состоянии технологий разработки программных продуктов; выработка умений в анализе исходную документацию и разработки документации по информационным системам и программному обеспечению; получение навыков оценки качества процессов создания программного обеспечения; выработка практических навыков проектирования программного обеспечения и расчета его надежности.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Технология программирования» относится к части, формируемой участниками образовательных отношений профессионального цикла Б1.В.03 основной образовательной программы подготовки бакалавров по направлению «09.03.01 «Информатика и вычислительная техника».

Перечень последующих дисциплин, для которых необходимы знания, умения и навыки, формируемые данной дисциплиной:

- Б1.В.01 Программирование компьютерной графики (ПК-6)
- Б1.В.05 Базы данных (ПК-3, ПК-6)
- Б1.В.13 Сопровождение программного обеспечения (ПК-6)
- Б1.В.17 Проектирование Web-приложений (ПК-6)

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины направлено на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОП ВО по данному направлению подготовки (специальности):

ПК-3 - Способен выявлять требования к базам данных, информационным системам и выполнять работы по их проектированию.

ПК-6 - Способен разрабатывать требования к программному обеспечению и проектировать его на основе современных технологий программирования.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы достижения компетенций	Результаты обучения
ПК-3. Способен выявлять требования к базам данных, информационным системам и выполнять работы по их проектированию	ПК-3.1. Выявляет требования к базам данных и информационным системам	<p>Знает: возможности информационных систем и баз данных; устройство и функционирование современных информационных систем; современные стандарты информационного взаимодействия систем; инструменты и методы анализа требований; характеристики различных систем обеспечения безопасности, влияющие на производительность информационных систем и баз данных.</p> <p>Умеет: анализировать исходную документацию и разрабатывать документацию по информационным системам и базам данных; настраивать параметры инструментов системы безопасности в соответствии с установленными критериями.</p> <p>Владеет: навыками анализа функциональных и нефункциональных требований к информационным системам и базам данных; навыками документирования и верификации требований к информационным системам и базам данных.</p>
	ПК-3.2. Проектирует базы данных и информационные системы	<p>Знает: языки программирования и работы с базами данных и информационными системами; инструменты и методы проектирования структур баз данных.</p> <p>Умеет: кодировать на языках программирования и верифицировать структуру программного кода.</p> <p>Владеет: инструментами и мето-</p>

		дами проектирования и дизайна информационных систем и баз данных; навыками выбора наиболее эффективных путей снижения нагрузки при обеспечении заданного уровня безопасности данных.
ПК-6. Способен разрабатывать требования к программному обеспечению и проектировать его на основе современных технологий программирования	ПК-6.1. Разрабатывает требования к программному обеспечению	<p>Знает: возможности современных и перспективных средств разработки программных продуктов, технических средств; методологии разработки программного обеспечения и технологии программирования; возможности существующей программно-технической архитектуры.</p> <p>Умеет: вырабатывать варианты реализации требований; проводить оценку и обоснование рекомендуемых решений; проводить анализ исполнения требований к программному обеспечению.</p> <p>Владеет: навыками анализа возможности, оценки времени и трудоемкости реализации требований к программному обеспечению</p>
	ПК-6.2. Проектирует программное обеспечение	<p>Знает: принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения.</p> <p>Умеет: применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.</p> <p>Владеет: навыками проектирования структур данных и программных интерфейсов, разработки и согласования технических спецификаций на программные компоненты и их взаимодействие.</p>

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Индекс	Наименование	Сессия 3										Итого за курс									
		Контроль					Академических часов					Академических часов					з.е.				
		Экз	КР	Всего	Кон такт.	Лек	Лаб	Пр	КРП	СР	Контроль	з.е.	Всего	Кон такт.	Лек	Лаб		Пр	КРП	СР	Контроль
Б1.В.03	Технология программирования	Экз	КР	252	24	8	12		4	147	9	5	252	24	8	12		4	147	9	5

ОБОЗНАЧЕНИЯ:

Виды промежуточной аттестации (виды контроля):

Экз - экзамен;
 ЗаО - зачет с оценкой;
 За – зачет;

Виды работ:

Контакт. – контактная работа обучающихся с преподавателем;
 Лек. – лекционные занятия;
 Лаб.– лабораторные работы;
 Пр. – практические занятия;
 КРП – курсовая работа (курсовой проект);
 РГР – расчетно-графическая работа (реферат);
 СР – самостоятельная работа студентов;
 з.е. – объем дисциплины в зачетных единицах.

Содержание дисциплины:

№	Наименование видов занятий и тематик, содержание
1	<p>лекционные занятия 4 шт. по 2 часа:</p> <p><i>1.1. Основы программирования в среде C#</i> Язык программирования C#. Структура программы. Операции уменьшения, увеличения, операции условия. Логические операции. Типы данных. Приведение типов. Массивы в C#. Многомерные массивы. Ступенчатые массивы. Функции класса System.Array. Операторы условия и выбора. Операторы циклов. Циклы for и foreach. Классы для работы со строками System.String и System.StringBuilder. Регулярные выражения. Классы в C#. Наследование. Интерфейсы, делегаты, события. Объявление делегатов. Использование делегатов. Простые делегаты. Групповые делегаты. Анонимные методы. Определение событий. Коллекции. Проблемы разработки сложных программных систем. Блочнo-иерархический подход к созданию сложных систем. Жизненный цикл и этапы разработки программного обеспечения. Эволюция моделей жизненного цикла программного обеспечения. Ускорение разработки программного обеспечения. Технология RAD. Оценка качества процессов создания программного обеспечения. Понятие технологичности программного обеспечения. Модули и их свойства. Нисходящая и восходящая разработка программного обеспечения. Структурное и «неструктурное» программирование. Средства описания структурных алгоритмов (псевдокоды, схемы алгоритмов, Flow-формы, диаграммы Насси-Шнейдермана). Стиль оформления программы. Эффективность и технологичность. Программирование «с защитой от ошибок». Сквозной структурный контроль.</p> <p><i>1.2. Классификация программного обеспечения</i> Классификация программных продуктов по функциональному признаку. Основные эксплуатационные требования к программным продуктам. Предпроектные исследования предметной области. Разработка технического задания. Определение технического задания. Принципиальные решения начальных этапов проектирования. Классификация моделей разрабатываемого программного обеспечения. Анализ требований и определение спецификаций при структурном подходе. Спецификации процессов. Словарь терминов.</p> <p><i>1.3. Проектирование программного обеспечения при структурном подходе к программированию</i> Диаграммы переходов состояний (SDT-диаграммы). Функциональные диаграммы (IDEF0). Диаграммы потоков данных (DFD). Диаграммы отношений компонентов данных: диаграммы Джексона и скобочные диаграммы Орра, сетевая модель данных (диаграмма «сущность-связь»). Структурная схема разрабатываемого программного обеспечения. Функциональная схема. Метод пошаговой детализации при составлении алгоритмов. Структурные карты Константайна. Структурные карты Джексона. Виды контроля качества разрабатываемого программного обеспечения. Ручной контроль программного обеспечения. Структурное тестирование программного обеспечения. Особенности структурного тестирования. Способ тестирования базового пути. Поточковый граф. Цикломатическая сложность. Шаги способа тестирования базового пути. Способы тестирования условий. Тестирование ветвей и операторов отношений. Способ тестирования потоков данных. Тестирование циклов. Простые циклы. Вложенные циклы. Объединенные циклы. Неструктурированные циклы. Особенности функционального тестирования. Разбиение на классы эквивалентности. Анализ граничных значений. Диаграммы причинно-следственных связей. Классификация ошибок. Методы отладки программного обеспечения. Методы и средства получения дополнительной информации. Общая методика отладки программного обеспечения. UML- стандартный язык описания разработки программных продуктов с использованием объектного подхода. Диаграммы вариантов использования. Описание вариантов использо-</p>

	<p>вания. Виды отношений между вариантами использования – ассоциация, расширение (extend), включение (include), обобщение. Диаграмма классов. Построение концептуальной модели предметной области. Класс: имя класса, атрибуты класса, операции. Отношения между классами – отношение зависимости, ассоциации, агрегации, композиции, обобщения. Интерфейсы. Объекты. Параметризованные классы (шаблоны).</p> <p><i>1.4 Разработка программных продуктов с использованием объектного подхода</i></p> <p>Диаграмма деятельности. Состояние действия. Переходы. Дорожки. Объекты. Диаграмма последовательностей. Линия жизни объекта. Фокус управления. Сообщения. Ветвление потока управления. Стереотипы сообщений. Временные ограничения на диаграммах последовательностей. Диаграмма кооперации. Объекты. Составные объекты. Связи. Сообщения. Диаграмма пакетов. Диаграммы состояний объекта. Состояние – имя состояния, список внутренних действий, начальное состояние, конечное состояние. Переход. Событие. Сторожевое условие. Выражение действия. Составное состояние и подсостояние. Последовательные подсостояния. Параллельные подсостояния. Историческое состояние. Сложные переходы. Диаграмма компонентов. Имя компонента. Виды компонентов. Интерфейсы. Зависимости. Диаграмма размещения. Узел. Соединения. Типы пользовательских интерфейсов и этапы их разработки. Психофизические особенности человека, связанные с восприятием, запоминанием и обработкой информации. Пользовательская и программная модели интерфейса. Классификации диалогов и общие принципы их разработки. Основные компоненты графических пользовательских интерфейсов. Реализация диалогов в графическом пользовательском интерфейсе. Пользовательские интерфейсы прямого манипулирования и их проектирование. Интеллектуальные элементы пользовательских интерфейсов. Граф диалога с пользователем. Разработка графа абстрактного диалога управляемого системой. Разработка графа абстрактного диалога управляемого пользователем. Разработка графа абстрактного диалога комбинированного типа. Общие положения стандарта. Классификация показателей качества программных средств. Методика оценки качества программных средств.</p>
2	<p>лабораторные работы 6 шт. по 2 часа:</p> <p><i>2.1. Линейные и циклические программы. Работа с массивами в C#</i></p> <p>Изучение основ языка C# и знакомство с элементами управления C#. Составление линейных и циклических программ. Работа с математическими функциями C#. Особенности работы с математическими функциями в C# (библиотека Math); операторы ветвления (if) и выбора (switch); операторы циклов (for, foreach, while и do...while). Особенности работы с массивами в языке C#, свойства и методы класса System.Array. Решение задач с одномерными, двумерными прямоугольными и двумерными ступенчатыми массивами.</p> <p><i>2.2. Работа со строками и регулярными выражениями</i></p> <p>Работа с элементами класса String. Работа со строками класса StringBuilder. Регулярные выражения. Оценка характеристик разработанных программ с помощью метрик Холстеда; метрик Джилба.</p> <p><i>2.3. Работа с файлами</i></p> <p>Оценка надежности программных средств с помощью модели Джелиински – Моранды, эвристической модели и модели Нельсона. Оценка степени отлаженности разрабатываемых приложений с помощью модели Миллса. Построение диаграммы переходов состояния, функциональной диаграммы. Построение структурной, функциональной схемы разрабатываемого программного обеспечения.</p> <p><i>2.4. Ручное тестирование программного обеспечения</i></p> <p>Структурное тестирование программного обеспечения. Функциональное тестирование программного обеспечения. Отладка программного обеспечения.</p> <p>Изучение методов тестирования программного обеспечения. Выполнение ручного тестирования программ. Выполнение структурного тестирования программ следующими мето-</p>

	<p>дами: тестирование базового пути; тестирование условий; тестирование циклов; тестирование потоков данных. Выполнение функционального тестирования программ следующими методами: разбиение на классы эквивалентности; анализ граничных значений; анализ причинно-следственных связей; предположение об ошибке. Выполнение отладки программного обеспечения методами индукции и методами дедукции. Сравнительная характеристика данных методов отладки программного обеспечения.</p> <p><i>2.5. Работа с классами в C#. Наследование. Разработка UML – диаграмм.</i> Цель работы: Создать класс. Каждый разрабатываемый класс должен, как правило, содержать следующие элементы: скрытые поля; конструкторы с параметрами и без параметров, методы, свойства. Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса. При возникновении ошибок должны выбрасываться исключения. В программе должна выполняться проверка всех разработанных элементов класса. Создать дочерний класс.</p> <p><i>2.6. Разработка пользовательских интерфейсов</i> Цель работы - разработать объектно-ориентированное, многооконное приложение с использованием интерфейсов, делегатов и событий, реализующее автоматизацию процессов в заданной предметной области. Разработать UML – диаграммы для созданного в предыдущей работе многооконного приложения. Обосновать тип пользовательского интерфейса и форму диалога. Разработать граф диалога пользователя. Разработать пользовательское меню.</p>
3	<p>практические занятия не предусмотрены в структуре дисциплины</p>
4	<p>курсовая работа <i>Проектирование, тестирование, отладка и оценка качества программных средств</i> Цель курсовой работы по дисциплине «Технология программирования» – закрепление соответствующего лекционного материала дисциплины, приобретение практических навыков проектирования программных систем с использованием структурного подхода, формирование компетенций, связанных с обеспечением требуемых технологических свойств разрабатываемого программного обеспечения, овладение современными технологиями проектирования приложений, методами расчета качества разрабатываемого программного обеспечения.</p> <p>Примеры тем курсовых работ:</p> <ol style="list-style-type: none"> 1. 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. 2. 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. 3. 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. 4. Система «Участковые». В системе должны поддерживаться режимы учета участковых (Ф.И.О., телефоны и т.д.) и их обслуживаемой территории (названия улиц, номера домов и т.д.), происшествий на обслуживаемой территории (дата происшествия, его вид, принятые меры и т.д.), поиска участкового по названию улицы, номера дома, анализа происшествий на обслуживаемой территории 5. Система «Утраченное оружие». В системе должны поддерживаться режимы учета утраченного оружия (похищенного, утерянного), поиска оружия по индивидуальному номеру,

	<p>формирования списка разыскиваемого оружия по видам (боевое, ручное, стрелковое, служебное, гражданское, газовое оружие и т.д.), анализ утраченного оружия по времени года, по районам города.</p> <p>6. 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.</p> <p>7. 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.</p> <p>8. «Федеральный розыск». В системе должны поддерживаться режимы учета лиц, объявленных в федеральный и межгосударственный розыск, поиска лиц по различным реквизитам (фамилия, имя, отчество, кличка, дата рождения, номер паспорта и т.д.), формирование списка лиц пропавших без вести.</p> <p>9. Система «Паспорт». В системе должны поддерживаться режимы ввода информации о гражданине, результатах его проверки по спец. учетам; поиска гражданина по реквизитам его паспорта; поиска паспорта по ФИО гражданина, формирование списка паспортов с истекшим сроком действия, проверка паспорта на его подлинность.</p> <p>10. Система «Штрафы за нарушение правил дорожного движения пешеходами». В системе должны поддерживаться режимы учета штрафов за нарушение правил дорожного движения, а также поиск неоплаченных штрафов.</p>
5	расчетно-графическая работа не предусмотрена в структуре дисциплины
6	<p>Самостоятельная работа студентов:</p> <p>6.1. 4 контрольных опроса после 1-й, 2-й, 3-й и 4-й лекций;</p> <p>6.2. Закрепление материала по тематике лекционных занятий: закрепление изучения материалов лекций 1.1-1.4 – основы программирования на языке высокого уровня; классификация программного обеспечения; проектирование программного обеспечения при структурном и объектно-ориентированном подходе к программированию; тестирование и отладка программных продуктов; оценка качества разработанных программных средств.</p> <p>6.3. Подготовка к экзамену по дисциплине (оценочные материалы приведены в разделе 6 настоящей РПД).</p>

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Таблица - Образовательные технологии, используемые при реализации различных видов учебной деятельности по дисциплине

№ п/п	Виды учебных занятий	Образовательные технологии
1	Лекции	Классическая (традиционная, информационная) лекция. Интерактивная лекция (лекция-визуализация). Индивидуальные и групповые консультации по дисциплине.
2	Лабораторная работа	Технология выполнения лабораторных заданий индивидуально. Технология выполнения лабораторных заданий в малой группе (в бригаде). Технология проблемного обучения на основе анализа результатов лабораторной работы: индивидуальный опрос, собеседование в малой группе (бригаде), обсуждение результатов командной работы, групповая дискуссия,

		метод «круглого стола», представление студентом или группой студентов (бригадой) результатов лабораторной работы в форме отчета и мультимедийной презентации. Проектная технология.
3	Консультации по курсовой работе (курсовому проекту)	Индивидуальные и групповые консультации. Информационно-коммуникационные технологии: технология взаимодействия со студентами в синхронном режиме связи — «offline»; технология взаимодействия со студентами в синхронном режиме связи — «online».
4	Самостоятельная работа студентов (внеаудиторная)	Информационно-коммуникационные технологии (доступ к ЭИОС филиала, к ЭБС филиала, доступ к информационно-методическим материалам по дисциплине).
5	Контроль (промежуточная аттестация: зачет или экзамен)	Технология письменного контроля, в том числе тестирование. Рейтинговая система контроля.

6. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ – ДЛЯ ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ДИСЦИПЛИНЫ

К промежуточной аттестации студентов по дисциплине могут привлекаться представители работодателей, преподаватели последующих дисциплин, заведующие кафедрами.

Оценка качества освоения дисциплины включает как текущий контроль успеваемости, так и промежуточную аттестацию.

Оценочные средства текущего контроля успеваемости:

Примеры вопросов к контрольному опросу после 1-й лекции:

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 Джелински – Моранды.
11. Опишите метод оценки надежности программных средств с помощью модели Нельсона.

12. Опишите метод оценки надежности программных средств с помощью эвристической модели.

Примеры вопросов к контрольному опросу после 2-й лекции:

1. Назовите основные эксплуатационные требования к программным продуктам. 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?
4. Составьте техническое задание на разработку «калькулятора» по типу, предлагаемого Windows.
5. Разработать программный модуль «Учет успеваемости стуков». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов 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 повре-

менной). Считается, что повременная оплата местных телефонных разговоров уже введена.

Примеры вопросов к контрольному опросу после 3-й лекции:

1. Определите понятие тестирования. Что такое тест? Поясните содержание процесса тестирования.
3. Что такое исчерпывающее тестирование? Какие задачи решает тестирование? Каких задач не решает тестирование? Какие принципы тестирования вы знаете? В чем их отличие друг от друга?
7. В чем состоит суть тестирования «черного ящика»? В чем состоит суть тестирования «белого ящика»? Каковы особенности тестирования «белого ящика»? Какие недостатки имеет тестирование «белого ящика»? Какие достоинства имеет тестирование «белого ящика»?
12. Дайте характеристику способа тестирования базового пути. Какие особенности имеет потоковый граф? Поясните понятие независимого пути.
15. Поясните понятие цикломатической сложности. Что такое базовое множество? Какие свойства имеет базовое множество? Какие способы вычисления цикломатической сложности вы знаете?
19. Поясните шаги способа тестирования базового пути. Поясните достоинства, недостатки и область применения способа тестирования базового пути.
21. Дайте общую характеристику способов тестирования условий. Какие типы ошибок в условиях вы знаете? Какие методики тестирования условий вы знаете?
24. Поясните суть способа тестирования ветвей и операторов отношений. Какие он имеет 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. что 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?

Примеры вопросов к контрольному опросу после 4-й лекции:

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 С#.

6. Что понимается под термином «конструктор», В чем состоит назначение конструктора, Каждый ли класс языка С# имеет конструктор, Какие умолчания для конструкторов приняты в языке С#? Приведите синтаксис конструктора класса в общем виде. Проиллюстрируйте его фрагментом программы на языке С#.

7. Что понимается под термином «деструктор», В чем состоит назначение деструктора? Приведите синтаксис деструктора класса в общем виде. Проиллюстрируйте его фрагментом программы на языке С#.

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 С#.

Примеры алгоритма самостоятельной работы по закреплению материала по тематике лекционных занятий:

В ходе изучения дисциплины «Технология программирования» студенты могут посещать аудиторские занятия (лекции, лабораторные занятия, консультации). Особенность изучения дисциплины состоит в выполнении комплекса лабораторных работ, главной задачей которого является получение навыков самостоятельной работы на компьютерах с использованием современных компьютерных программ, предназначенных для решения определенного круга профессиональных задач.

Важное место в овладении тем данной дисциплины отводится самостоятельной работе, при этом во время аудиторных занятий могут быть рассмотрены и проработаны наиболее важные и трудные вопросы по той или иной теме дисциплины, а более легкие вопросы могут быть изучены студентами самостоятельно.

Методика закрепления материалов лекционных занятий 1.1-1.4:

Закрепление знаний в области данной дисциплины, приобретение практических навыков проектирования программных систем с использованием структурного и объектно-ориентированного подходов осуществляется путем разработки программных средств по заданной предметной области:

1. Анализ требований на разработку.

Формирование требований к разрабатываемой системе; обследование объекта и обоснование необходимости создания программных средств; формирование требований пользователя к разрабатываемой системе; оформление отчета о выполнении работ и заявки на разработку программных средств; разработка концепции программного обеспечения; изучение объекта проектирования; Проведение необходимых исследовательских работ; разработка вариантов концепции и выбор варианта концепции программных средств, удовлетворяющего требованиям пользователей; оформление отчета о проделанной работе.

2. Разработка и утверждение технического задания.

Разработка и утверждение технического задания включает в себя: определение требований к программе; разработку технического обоснования разработки программы; определение стадий, этапов и сроков разработки программы и документации на нее; выбор языков программирования; определение необходимости проведения исследовательских работ на последующих стадиях; согласование и утверждение технического задания.

Результатом данной фазы является общее описание системы, включающее в себя требования к программе и требования к надежности программы. Требования к программному обеспечению включают: описание входных данных (корректных и ошибочных); роли при использовании программного обеспечения, и их интерфейсы (средств общения с пользователями); упрощения, предположения и допущения по отношению к программам; описание выходных данных; требова-

ния к надежности функционирования программы. Определяются ошибки, которые необходимо выявлять, и сообщения, которые желательно выдавать пользователю при наличии ошибок. Перечисляются все особые ситуации, требующие дополнительного учета и специального рассмотрения.

3. Проектирование программного обеспечения.

Проектирование включает в себя два этапа: эскизный проект и технический проект. Эскизный проект заключается в предварительной разработке структуры входных и выходных данных и общего описания алгоритма решения задачи. При разработке технического проекта структуры входных и выходных данных уточняются и определяются их формы представления. Уточняется алгоритм решения задачи, определяется семантика и синтаксис языка программирования и разрабатывается структура программы. Оба этапа сопровождаются пояснительной запиской и технико-экономическим обоснованием. В соответствии с технологией нисходящего структурного программирования программный комплекс разбивается на небольшие части – программные модули. Каждая отдельная подзадача должна быть относительно независимой и представлять собой некоторый законченный модуль программы. По итогу этапа должны быть спроектированы: словарь терминов, функциональные диаграммы IDEF0 или диаграммы потоков данных (DFD). Не менее трех уровней иерархии диаграмм (лучше больше). Оценка качества проектирования (коэффициент декомпозиции, коэффициент сбалансированности); диаграммы переходов состояний STD; схемы алгоритмов; диаграммы Джексона, структурные карты Константайна.

4. Реализация рабочего проекта.

Рабочий проект включает в себя разработку программы и программной документации, а также испытание программы. Этап кодирования алгоритмов заключается в переводе алгоритмов, разработанных для каждого программного модуля, в программы на конкретном языке программирования. Кодирование должно быть простым. Изощренное программирование может обойтись слишком дорого при отладке и модификации программы. Необычное кодирование (например, использование скрытых возможностей машины) часто препятствует отладке программы и, конечно, затрудняет ее использование другими программистами. Входные форматы должны быть разработаны с учетом максимального удобства для пользователя и минимальной возможности ошибок. Порядок переменных и форматы данных, привычные для пользователя, помогут избежать ошибок и облегчат использование программ. Форматы выходных данных могут сильно различаться. Иногда даются четкие инструкции, и выходные данные подгоняются под определенный стандарт. Результаты расчета должны быть удобочитаемыми и понятными для непрограммиста. По итогу этапа должны быть разработаны: структурные и функциональные схемы; описание разработанных процедур и функций (название, входные данные, выходные данные, ограничения); структурное тестирование (тестирование базового пути, тестирование условий, тестирование циклов), функциональное тестирование (разбиение на классы эквивалентности, анализ граничных значений, анализ причинно-следственных связей); реализация пользовательского интерфейса (построение графа диалога).

В заключении поводится оценка качества разработанного программного обеспечения.

Оценочные средства для промежуточной аттестации:

Примеры вопросов к экзамену по дисциплине:

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. CASE-средства и технологии. Определения, структура, классификация, примеры.
30. Структурное тестирование. Тестирование базового пути
31. Структурное тестирование. Тестирование условий
32. Структурное тестирование. Тестирование циклов
33. Структурное тестирование. Тестирование потоков данных
34. Функциональное тестирование. Разбиение на классы эквивалентности и анализ граничных значений
35. Функциональное тестирование. Анализ причинно-следственных связей
36. Классификация ошибок
37. Методы отладки программного обеспечения
38. Разработка пользовательского интерфейса. Классификация диалогов и общие принципы их работы
39. Разработка пользовательского интерфейса. Граф диалога с пользователем
40. 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 Равнобочная 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.

В филиале используется система с традиционной шкалой оценок – "отлично", "хорошо", "удовлетворительно", "неудовлетворительно", "зачтено", "не зачтено" (далее - пятибалльная система).

Форма промежуточной аттестации по настоящей дисциплине – **экзамен на 3 сессии 4 курса.**

Применяемые критерии оценивания по дисциплинам (в соответствии с инструктивным письмом НИУ МЭИ от 14 мая 2012 года № И-23):

Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
«отлично»/ «зачтено (отлично)»/ «зачтено»	Выставляется обучающемуся, обнаружившему всестороннее, систематическое и глубокое знание материалов изученной дисциплины, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной рабочей программой дисциплины; проявившему творческие способности в понимании, изложении и использовании материалов изученной дисциплины, безупречно ответившему не только на вопросы билета, но и на дополнительные вопросы в рамках рабочей программы дисциплины, правильно выполнившему практическое задание. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущего контроля. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «эта-

Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
	лонный».
«хорошо»/ «зачтено (хорошо)»/ «зачтено»	Выставляется обучающемуся, обнаружившему полное знание материала изученной дисциплины, успешно выполняющему предусмотренные задания, усвоившему основную литературу, рекомендованную рабочей программой дисциплины; показавшему систематический характер знаний по дисциплине, ответившему на все вопросы билета, правильно выполнивший практическое задание, но допустивший при этом непринципиальные ошибки. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущего контроля. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «продвинутой».
«удовлетворительно»/ «зачтено (удовлетворительно)»/ «зачтено»	Выставляется обучающемуся, обнаружившему знание материала изученной дисциплины в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющемуся с выполнением заданий, знакомому с основной литературой, рекомендованной рабочей программой дисциплины; допустившему погрешность в ответе на теоретические вопросы и/или при выполнении практических заданий, но обладающему необходимыми знаниями для их устранения под руководством преподавателя, либо неправильно выполнившего практическое задание, но по указанию преподавателя выполнившего другие практические задания из того же раздела дисциплины. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «пороговой».
«неудовлетворительно»/ не зачтено	Выставляется обучающемуся, обнаружившему серьезные пробелы в знаниях основного материала изученной дисциплины, допустившему принципиальные ошибки в выполнении заданий, не ответившему на все вопросы билета и дополнительные вопросы и неправильно выполнившего практическое задание (неправильное выполнение только практического задания не является однозначной причиной для выставления оценки «неудовлетворительно»). Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущего контроля. Компетенции на уровне «пороговой», закреплённые за дисциплиной, не сформированы.

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Учебное и учебно-лабораторное оборудование

Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная: специализированной мебелью; доской аудиторной; демонстрационным оборудованием: персональным компьютером (ноутбуком); переносным (стационарным) проектором.

Учебная аудитория для лабораторных работ, выполняемых в компьютерном классе, оснащенная: специализированной мебелью; доской аудиторной; персональными компьютерами с подключением к сети "Интернет" и доступом в ЭИОС филиала.

Для самостоятельной работы обучающихся по дисциплине используется помещение для самостоятельной работы обучающихся, оснащенное: специализированной мебелью; доской аудиторной; персональным компьютерами с подключением к сети "Интернет" и доступом в ЭИОС филиала.

Программное обеспечение (Операционная система OS Windows, офисный пакет Microsoft Office, Microsoft Visual Studio 2019 Community, Draw.io)

8. ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

для слепых и слабовидящих:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
- для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
- письменные задания оформляются увеличенным шрифтом;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

для глухих и слабослышащих:

- лекции оформляются в виде электронного документа;
- письменные задания выполняются на компьютере в письменной форме;
- экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

для лиц с нарушениями опорно-двигательного аппарата:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере;
- используется специальная учебная аудитория для лиц с ЛОВЗ – ауд. 106 главного учебного корпуса по адресу 214013, г. Смоленск, Энергетический пр-д, д.1, здание энергетического института (основной корпус).

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены филиалом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

для слепых и слабовидящих:

- в печатной форме увеличенным шрифтом;
- в форме электронного документа;

- в форме аудиофайла.

для глухих и слабослышащих:

- в печатной форме;

- в форме электронного документа.

для обучающихся с нарушениями опорно-двигательного аппарата:

- в печатной форме;

- в форме электронного документа;

- в форме аудиофайла.

9. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература.

1. Иванова Г.С. Технология программирования: учебник / Г.С. Иванова – М. КНО-РУС, 2017. -336 с.

2. Кирсяев А.Н. Теория и технология программирования. Программное обеспечение вычислительной математики [Электронный ресурс] : учебное пособие / А.Н. Кирсяев. — Электрон. дан. — Санкт-Петербург : СПбГПУ, 2017. — 104 с. — Режим доступа: по подписке. — URL: <https://e.lanbook.com/book/105484>.

3. Мишова В.В. Технологии программирования : практикум / В.В. Мишова ; Министерство культуры Российской Федерации, Кемеровский государственный институт культуры, Институт информационных и библиотечных технологий. – Кемерово : Кемеровский государственный институт культуры (КемГИК), 2016. – 87 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=472686>. – Библиогр.: с. 84. – ISBN 978-5-8154-0360-4. – Текст: электронный.

Дополнительная литература.

1. Кручинин В.В. Технологии программирования [Электронный ресурс] : учебное пособие / В.В. Кручинин. — Электрон. дан. — Москва : ТУСУР, 2013. — 271 с. — Режим доступа: <https://e.lanbook.com/book/110371>.

2. Горелов С.В. Современные технологии программирования: разработка Windows-приложений на языке С#: учебник для студентов, обучающихся по дисциплине «Современные технологии программирования», направление «Прикладная информатика»: в 2 томах : [16+] / С.В. Горелов ; под науч. ред. П.Б. Лукьянова ; Финансовый университет при Правительстве Российской Федерации. – Москва : Прометей, 2019. – Том 1. – 363 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=576037>. – Библиогр. в кн. – ISBN 978-5-907100-09-1. – Текст: электронный.

Список авторских методических разработок.

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»: методические указания / А.И. Гаврилов, В.В. Малахов, И.В. Малашенкова, Е.А. Панкратова, О.В. Семенова – Смоленск: РИО филиала МЭИ в г.Смоленске, 2014. – 110 с.

Панкратова Е.А. Тестирование программного обеспечения [Текст]: методические рекомендации / Е.А. Панкратова, О.В. Семенова - Смоленск: РИО филиала МЭИ в г. Смоленске, 2011. – 24 с.

Федулов Я.А. Комплект мультимедийных презентаций к лекциям по дисциплине «Технология программирования» (расположен в ЭИОС филиала и передается обучающимся на 1-й лекции для подготовки к лекциям и самостоятельного изучения дисциплины).

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Но- мер изме- мене- ния	Номера страниц				Всего стра- ниц в доку- менте	Наименование и № документа, вводящего изменения	Подпись, Ф.И.О. внесшего измене- ния в данный эк- земпляр	Дата внесения из- менения в данный эк- земпляр	Дата введения из- менения
	изме- нен- ных	заме- нен- ных	но- вых	анну- лиро- ван- ных					
1	2	3	4	5	6	7	8	9	10