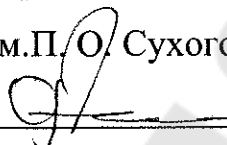


Учреждение образования «Гомельский государственный технический университет имени П.О. Сухого»

УТВЕРЖДАЮ
Первый проректор ГГТУ

им. П.О. Сухого

 О.Д. Асенчик

09. 12. 2015

Регистрационный № УД- 45 - 07 /уч.

ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ СИСТЕМ УПРАВЛЕНИЯ

Учебная программа учреждения высшего образования
по учебной дисциплине для специальности

1-53 01 07 «Информационные технологии и
управление в технических системах»

Учебная программа составлена на основе:

- типовой учебной программы по учебной дисциплине «Технология разработки программного обеспечения систем управления» № 1.1263, утвержденной 15.09.2015;

- учебных планов учреждения образования «Гомельский государственный технический университет имени П.О. Сухого» специальности 1-53 01 07 «Информационные технологии :: управление в технических системах», регистрационные №№ 1 53-1-38/уч. 17.04.2014; 1 53-1-04/уч. 12.02.2015.

СОСТАВИТЕЛИ:

Э.М. Виноградов, доцент кафедры «Промышленная электроника» учреждения образования «Гомельский государственный технический университет имени П.О. Сухого», кандидат технических наук, доцент;

А.В. Сахарук, ассистент кафедры «Промышленная электроника» учреждения образования «Гомельский государственный технический университет имени П.О. Сухого».

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой «Промышленная электроника» учреждения образования «Гомельский государственный технический университет имени П.О. Сухого» (протокол №3 от 15.10. 2015 г.);

Научно-методическим советом факультета автоматизированных и информационных систем учреждения образования «Гомельский государственный технический университет имени П.О. Сухого» (протокол № 4 от 30.11. 2015 г.);

Удф - 05-07/уч.

Научно-методическим советом учреждения образования «Гомельский государственный технический университет имени П.О. Сухого» (протокол №2 от 08.12. 2015 г.).

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Введение.

Изучение учебной дисциплины «Технология разработки программного обеспечения систем управления» осуществляется в соответствии с требованиями к формированию академических, социально-личностных и профессиональных компетенций специалиста в сфере программирования.

Цели и задачи учебной дисциплины.

Цель учебной дисциплины – ознакомить студентов с основными технологиями, применяемыми в настоящее время для разработки программного обеспечения современных автоматизированных и автоматических систем, научить студентов способам использования обширного инструментария разработчика программного обеспечения.

Задачи дисциплины:

- подготовка специалистов в области информационных технологий и управления, имеющих достаточный объем знаний и практические навыки создания управляющих программ и программного обеспечения для автоматических и автоматизированных систем.
- приобретение знаний в области разработки программного обеспечения для систем управления;
- формирование навыков разработки управляющих программ и программных интерфейсов для автоматических и автоматизированных систем,
- изучение принципов объектно-ориентированного программирования и создание приложений под различные операционные системы,
- овладение методами создания программного обеспечения для систем управления.

Место учебной дисциплины в системе подготовки специалистов, связи с другими учебными дисциплинами.

Учебная дисциплина «Технология разработки программного обеспечения систем управления» входит в состав государственного компонента цикла общепрофессиональных и специальных дисциплин. Данная дисциплина связана с отдельными разделами таких учебных дисциплин, как «Основы алгоритмизации и программирования», «Основы защиты информации», «Основы теории систем» и «Теория передачи информации».

Требования к освоению учебной дисциплины.

После изучения дисциплины «Технология разработки программного обеспечения систем управления» подготавливаемый специалист должен соответствовать следующим требованиям к его компетентности:

академические компетенции:

- АК-1. Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.

- АК-2. Владеть системным и сравнительным анализом.
- АК-3. Владеть исследовательскими навыками.
- АК-4. Уметь работать самостоятельно.
- АК-6. Владеть междисциплинарным подходом при решении проблем.
- АК-10. Использовать основные законы естественнонаучных дисциплин в профессиональной деятельности.
- АК-12. Владеть основными методами защиты производственного персонала и населения от возможных последствий аварий, катастроф, стихийных бедствий.

социально-личностные компетенции:

- СЛК-3. Обладать способностью к межличностным коммуникациям.
- СЛК-5. Быть способным к критике и самокритике.
- СЛК-6. Уметь работать в команде.

профессиональные компетенции:

- ПК-1. Осваивать современные и разрабатывать перспективные системы автоматического и автоматизированного управления технологическими процессами преобразования энергии, вещества и информации.
- ПК-2. Разрабатывать алгоритмическое обеспечение для систем автоматического управления технологическими процессами и подвижными объектами.
- ПК-4. Разрабатывать, изготавливать и эксплуатировать электронные компоненты систем автоматического контроля и регулирования.
- ПК-6. Обеспечивать информатизацию управляющих объектов крупных технических, организационных и экономических комплексов, включающих управляющий персонал.
- ПК-7. Разрабатывать, тестировать и отлаживать программные средства для локальных микропроцессорных систем контроля, регулирования и управления.
- ПК-11. Анализировать и оценивать собранные данные.
- ПК-15. Владеть современными средствами инфокоммуникаций.

В результате освоения содержания учебной дисциплины «Технология разработки программного обеспечения систем управления» студент должен:

знать:

- современные методы и средства создания программного обеспечения систем управления;
- специализированные библиотеки разработчика программного обеспечения;
- основные приемы и подходы к созданию сложных программных комплексов;

уметь:

- применять на практике методологию объектно-ориентированного программирования;
- создавать программное обеспечение систем управления под операционные системы семейства Windows и Linux;

владеть:

- технологией создания программного интерфейса систем управления для ОС Windows и Linux;
- технологией объектно-ориентированного программирования.

Программа учебной дисциплины рассчитана на 324 учебных часа, аудиторных – 133 часа. Трудоемкость учебной дисциплины составляет 9 зачетных единиц.

Форма получения высшего образования – дневная.

Распределение аудиторного времени по видам занятия, курсам и семестрам:

Курс – 2,3

Семестры – 4,5

Лекции – 67 часов

Лабораторные занятия – 66 часов

Всего аудиторных часов – 133 ч.

Форма текущей аттестации – экзамены 4,5 семестры

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Интегрированная среда разработки Qt Creator. Особенности языков C++, C#

Тема 1.1. Предмет дисциплины и ее задачи

Структура, содержание дисциплины, ее связь с другими дисциплинами учебного плана. Роль объектно-ориентированного программирования, проектирования и анализа при создании систем управления.

Тема 1.2. Компиляторы gcc, g++

Работа с консольными приложениями. Создание нового проекта. Добавление к проекту файлов с исходным кодом. Компиляция, компоновка и выполнение проекта. Конфигурация проекта.

Тема 1.3. Особенности и синтаксис языков программирования

Комментарии. Соккрытие имен и унарная операция привязки. Операции new и delete для работы с динамической памятью. Особенности создания многомерных динамических массивов. Тип данных «ссылки»: псевдонимы имен и передача ссылок в функции в качестве аргументов. Ввод/вывод потоками. Передача аргументов функции по умолчанию. Механизм перегрузки функций (раннее связывание).

Раздел 2. Введение в ООП, понятие класса и объекта. Классы, дополнительные возможности методов классов

Тема 2.1. Основные принципы объектно-ориентированного программирования

Класс. Объект. Инкапсуляция. Внешние и внутренние проявления класса, абстрагирование. Ограничение доступа. Модульность. Иерархия классов и наследование. Полиморфизм, достоинства и недостатки объектно-ориентированного программирования.

Тема 2.2. Основной элемент ООП – класс. Основные компоненты, объекты

Характеристика элементов данных класса. Характеристика элементов-функций класса. Операция привязки. Встроенные функции. Полное имя компонентов класса. Указатель “this”. Методы-конструкторы объектов. Конструктор копирования.

Тема 2.3. Методы класса, способы применения, особенности

Дружественные (friend) функции класса. Способы передачи аргументов в функции-друзья класса. Методы const, не изменяющие объекты класса. Статические методы и данные. Особенности работы со статическими элементами класса.

Раздел 3. Наследование. Виртуальные функции. Переопределение операций

Тема 3.1. Правила и принципы наследования. Производные и базовые классы

Правила наследования. Управление уровнем доступа к элементам класса. Множественное наследование. Последовательность создания и уничтожения подобъектов. Инициализация объектов при наследовании. Указатели на базовый и производные классы. Виртуальный базовый класс. Конструктор во множественном наследовании.

Тема 3.2. Виртуальные функции. Позднее связывание

Механизм переопределения метода класса (позднее связывание). Виртуальные методы. Виртуальные деструкторы.

Тема 3.3. Переопределение операций

Общие положения и определения. Перегрузка методами и функциями-друзьями.

Раздел 4. Шаблоны (параметризованные типы). Другие возможности стандартных библиотек

Тема 4.1. Шаблоны функций, шаблоны классов

Требования к фактическим параметрам шаблона. Отождествление типов аргументов. Шаблоны классов. Шаблоны классов: не только для типов. Наследование в шаблонах классов.

Тема 4.2. Другие возможности стандартных библиотек

Указатели на компоненты класса. Понятие абстрактного класса. “Чистые” виртуальные функции. Шаблоны функций и классов. Особенности перегрузки шаблонов.

Раздел 5. Библиотека потоков данных. Обработка исключительных ситуаций

Тема 5.1. Библиотека потоков данных

Простое внесение. Выражение извлечения. Создание собственных функции внесения и извлечения. Функции библиотек разных языков программирования. Манипуляторы ввода-вывода. Файловые и строковые потоки.

Тема 5.2. Исключительные ситуации, обработка, основные операции

Операции try, catch, throw. Синтаксис операций, примеры использования.

Раздел 6. Разработка графического интерфейса

Тема 6.1. Особенности построения графического интерфейса с использованием библиотек Qt

Базовые классы Qjbest и QWidget. Система сигналов и слотов. Наследование от QWidget. Создание собственных компонентов на основе базовых классов.

Тема 6.2. Стандартные библиотеки и проектирование основных интерфейсов приложения

Преимущества использования стандартных библиотек. Иерархия классов стандартных библиотек. Описание классов. Понятие объекта приложения.

Тема 6.3. Аппаратно-независимая графика, файловая система и ресурсы приложения

Основные принципы работы и эффективное использование файловой системы. Сохранение и восстановление состояния объектов. Применение методов классов стандартных библиотек. Обеспечение сериализации данных приложения системы управления. Оптимизация вывода графики на экран. Графическое устройство и его контекст. Реализация принципа эффективной перерисовки рабочей области приложения. Главное меню приложения. Управление созданием, загрузкой и заменой меню. Динамически изменяющееся меню. Панель управления приложением. Ресурсы, форма и режимы работы панели управления. Расширение панели управления. Панель состояния работы приложения. Изменение характеристик индикаторов. Использование дополнительных возможностей панели состояния.

Тема 6.4. Поточковая многозадачность

Многозадачность, основанная на процессах и потоках. Потоки стандартных библиотек. Интерфейсные и рабочие потоки приложения. Создание, запуск, остановка и возобновление выполнения потоков. Организация управления приоритетами потоков. Синхронизация потоков. Объекты синхронизации и классы стандартных библиотек. Механизм синхронизации. Общая концепция управления доступом к разделяемым ресурсам приложения. Синхронизация на основе семафоров. Работа с объектами событий. Использование критических секций.

Тема 6.5. Приложение, основанное на диалоге

Мастера проектов. Создание шаблона приложения. Главный класс приложения. Класс диалоговой панели.

Тема 6.6. Диалоговые панели

Мастера проектов. Модальная диалоговая панель. Диалоговая панель - главное окно приложения. Немодальная диалоговая панель.

Раздел 7. Динамически подключаемые библиотеки

Тема 7.1. Создание динамической библиотеки на основе стандартных библиотек. Создание приложения, обращающегося к dll

Основные понятия о библиотеке динамической загрузки (DLL). Технология функционирования DLL. Различные типы DLL. Динамическая загрузка и выгрузка DLL. Создание DLL с использованием стандартных библиотек. Экспорт класса. Экспортирование объектов и переменных. Экспорт функции.

Тема 7.2. Эффективное использование библиотек динамической загрузки

Использование DLL в приложении. Использование DLL с библиотекой импорта. Использование DLL без библиотеки импорта. Эффективная загрузка DLL. Загрузка динамических расширений стандартных библиотек.

Раздел 8. Технология Qt Quick

Тема 8.1. Применение визуальных элементов, фиксаторов и пользовательский ввод

Общие положения Qt Quick. Язык описания графического интерфейса QML. Базовые элементы. Управление размещением элементов на форме. Компоненты для ввода и вывода информации .

Тема 8.2. Создание собственных элементов QML

Создание собственных графических компонентов. Компоновка элементов в пользовательские библиотеки.

Раздел 9. Поддержка баз данных

Тема 9.1. Доступ к данным БД

Поддержка СУБД: Oracle, Microsoft SQL Server, Sybase Adaptive Server, IBM DB2, PostgreSQL, MySQL и ODBC-совместимые базы данных.

Тема 9.2. Поддержка баз данных в стандартных библиотеках

Классы для работы с базами данных. Ошибки при операциях с базами данных. Создание приложения, работающего с базой данных.

Тема 9.3. Использование языка SQL в библиотеке Qt

Слой драйверов: QSqlDriver, QSqlDriverCreator, QSqlDriverCreatorBase, QSqlDriverPlugin и QSqlResult.

Слой SQL API: QSqlDatabase, QSqlQuery, QSqlError, QSqlField, QSqlIndex и QSqlRecord

Раздел 10. Разработка служб. Поддержка сети

Тема 10.1. Применение стандартных протоколов HTTP и FTP

Описание протоколов. Авторизация. Применение стандартных библиотек для работы с ними.

Тема 10.2. Сокетные соединения. TCP и UDP

Основы работы с SOCKET. Технология клиент-сервер. Поддержка сокетного соединения в Qt

Тема 10.3. Разработка служб

Понятие системной службы. Особенности разработки служб для Windows (Windows Service Applications). Особенности разработки служб (deamons) для Linux. Интеграция службы в операционную систему.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ
(дневная форма получения образования)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСП	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
Четвертый семестр								
1	Интегрированная среда разработки Qt Creator. Особенности языков C++, C#	4			3			
1.1	Предмет дисциплины и ее задачи	1						Опрос
1.2	Компиляторы gcc, g++	1			1			Защита ЛР
1.3	Особенности и синтаксис языков программирования	2			2			Защита ЛР
2	Введение в ООП, понятие класса и объекта. Классы, дополнительные возможности методов классов	12			6			
2.1	Основные принципы объектно-ориентированного программирования	4			2			Опрос
2.2	Основной элемент ООП – класс. Основные компоненты, объекты	4			2			Защита ЛР
2.3	Методы класса, способы применения, особенности	4			2			Защита ЛР
3	Наследование. Виртуальные функции. Переопределение операций	10			6			
3.1	Правила и принципы наследования. Производные и базовые классы	4			2			Опрос
3.2	Виртуальные функции. Позднее связывание	4			2			Опрос Защита ЛР
3.3	Переопределение операций	2			2			Опрос Защита ЛР
4	Шаблоны (параметризованные типы) другие возможности стандартных библиотек	4			2			
4.1	Шаблоны функций, шаблоны классов	2			2			Защита ЛР
4.2	Другие возможности стандартных библиотек	2						Опрос
5	Библиотека потоков данных. Обработка исключительных ситуаций	4			4			
5.1	Библиотека потоков данных	2			2			Опрос

5.2	Исключительные ситуации, обработка, основные операции	2		2			Защита ЛР
6	Разработка графического интерфейса	17		12			
6.1	Особенности построения графического интерфейса с использованием библиотек Qt	2		2			Опрос
6.2	Стандартные библиотеки и проектирование основных интерфейсов приложения	4		2			Защита ЛР
6.3	Аппаратно-независимая графика, файловая система и ресурсы приложения	4		2			Защита ЛР
6.4	Потоковая многозадачность	2		2			Опрос
6.5	Приложение, основанное на диалоге	2		2			Защита ЛР
6.6	Диалоговые панели	3		2			
	Текущая аттестация						Экзамен
Пятый семестр							
7	Динамически подключаемые библиотеки	4		8			
7.1	Создание динамической библиотеки на основе стандартных библиотек. Создание приложения, обращающегося к dll	2		4			Опрос
7.2	Эффективное использование библиотек динамической загрузки	2		4			Защита ЛР
8	Технология Qt Quick	3		6			
8.1	Применение визуальных элементов, фиксаторов и пользовательский ввод	2		2			Защита ЛР
8.2	Создание собственных элементов QML	1		4			Опрос
9	Поддержка баз данных	5		8			
9.1	Доступ к данным БД	2		2			Защита ЛР
9.2	Поддержка баз данных в стандартных библиотеках	2		4			Опрос
9.3	Использование языка SQL в библиотеке Qt	1		2			Защита ЛР
10	Разработка служб. Поддержка сети	4		11			
10.1	Применение стандартных протоколов HTTP и FTP	1		3			Защита ЛР
10.2	Сокетные соединения. TCP и UDP	1		4			Опрос
10.3	Разработка служб	2		4			Защита ЛР
	Текущая аттестация						Экзамен
	Итого	67		66			

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Лафоре, Роберт. Объектно-ориентированное программирование в C++/ Роберт Лафоре. - СПб. : Питер, 2014.
2. М. Саммерфилд Qt профессиональное программирования, разработка кроссплатформенных приложений на C++ / Символ 2011
3. А.Н. Боровский. Qt4.7 Практическое программирование на C++/БВХ-Петербург 2012
4. В.В. Лаптев. C++ объектно ориентированное программирование/Питер 2008
5. Д. Кьюу. Объектно ориентированное программирование/Питер 2005
6. Р. Лав. LINUX Системное программирование/Питер 2008

Дополнительная литература

7. Рудаков, А. В. Технология разработки программных продуктов. Практикум / А. В. Рудаков А. В., Г. Н. Федорова. - М.: Академия - Москва, 2010.
8. Богуславский, А. С ++ и компьютерная графика. Лекции и практикум по программированию на C ++/ А.С. Богуславский - М.: Компьютер Пресс, 2003.
9. Р. Лав Ядро Linux, описание процесса разработки 3-е издание / Вильямс 2015.
10. И.А. Волкова Основы объектно-ориентированного программирования. Язык программирования C++/ А.В. Иванов, Л.Е. Карпов. – М.: МГУ, 2011.
11. М. Шлее, Qt 5.3. Профессиональное программирование на C++/ – М.: БХВ-Петербург, 2015.
12. С. Мейерс Эффективное использование STL / Gbntb 2001

Список литературы сверен АИ (Тимова А.В.)

Перечень компьютерных программ, наглядных пособий, методических материалов и технических средств обучения

1. GIT – Комплексная система контроля версий.
2. Qt 5.5.1 for Windows 32-bit (MinGW 4.9.2) –Набор необходимых инструментов для разработки программного обеспечения.
3. OracleVMVirtualBox– Виртуальная машина с поддержкой различных операционных систем.

Примерный перечень тем лабораторных занятий

1. Компилятор gcc и интегрированная среда Qt Creator
2. Работа с динамической памятью, указателями на массивы данных, передача указателя на массив в качестве аргумента в функцию.
3. Использование классов, встроенных методов, доступ к классу через объект, указатель на объект и через динамическое выделение памяти под класс.
4. Использование методов классов: конструктора, деструктора и дружественной функции.
5. Наследование классов, механизм виртуальных функций, переопределение операций.
6. Программирование шаблонов функций и классов.
7. Файлового ввод - вывод с применением файловых потоков.
8. Функций обработки исключительных ситуаций.
9. Основы построения графического интерфейса. Базовый класс QWidget. Система сигналов и слотов.
10. Управление компоновкой элементов на форме.
11. Работа с элементами отображения.
12. Элементы ввода, управления и выбора.
13. Управление событиями.
14. Компоненты для работы с мультимедиа.
15. Многопоточность в приложении. Синхронизация потоков.
16. Работа с диалогами. Класс QDialog.
17. Построения основного окна приложения на основе QMainWindow.
18. Разработка и использование динамических библиотек.
19. Основы QML. Построение графического интерфейса.
20. Разработка собственных компонентов на основе QML.
21. Разработка приложений с поддержкой БД.
22. Работа с сетями.
23. Разработка системных служб (часть 1).
24. Разработка системных служб (часть 2).

Технологии обучения

Для организации процесса изучения учебной дисциплины «Технология разработки программного обеспечения систем управления» привлечены традиционные и инновационные образовательные технологии, ориентированные на формирование навыков самостоятельного и группового решения поставленных задач.

Лабораторные занятия проводятся с использованием персональных компьютеров. Контроль знаний проводится в ходе защиты лабораторных работ.

Организация самостоятельной работы студентов

Самостоятельная работа студентов организована в соответствии с Положением о самостоятельной работе студентов учреждения образования «Гомельский государственный технический университет имени П.О. Сухого» № 33, утвержденного ректором университета 14.10.2014.

Основными целями ее осуществления являются: активизация учебно-познавательной деятельности и формирование у студентов умений и навыков самостоятельного приобретения и практического применения знаний в области экономических и правовых аспектов предпринимательской деятельности в сфере промышленной электроники.

С учетом специфики и содержания учебной дисциплины «Технология разработки программного обеспечения систем управления» предполагается использование следующих форм самостоятельной работы студентов:

- контролируемая самостоятельная работа (проведение исследований необходимых для выполнения лабораторных работ в аудитории под контролем преподавателя);
- собственно самостоятельная работа (подготовка к рубежному контролю знаний и текущей аттестации (экзамену), организованная студентом самостоятельно).

Для организации эффективной самостоятельной работы студентов используется учебно-методическое обеспечение дисциплины, включающее современные информационные ресурсы и технологии (электронный курс дисциплины).

Средства диагностики результатов учебной деятельности

Процедура диагностики результатов учебной деятельности студентов разработана и организована в соответствии с Образовательным стандартом высшего образования первой ступени. Ее компоненты представлены:

- требованиями к осуществлению диагностики (определение объекта диагностики, измерение степени соответствия учебных достижений студента требованиям Образовательного стандарта ОСВО 1-53 01 07-2013, оценивание результатов измерения на основе принятой шкалы оценок);
- шкалой оценок (оценка промежуточных и итоговых (экзаменационных) достижений студента производится по десятибалльной шкале в зависимости от количества и качества выполненных заданий, предусмотренных планом);
- критериями оценок, разработанными учреждением образования;
- инструментарием диагностики (выполнение и защита лабораторных работ, разработка ПО (ПК-1, ПК-2, ПК-4, ПК-6, ПК-7, ПК-11, ПК-15)





Для диагностики соответствия учебных достижений студента предъявляемым требованиям используются типовые индивидуальные и лаборатор-

ные и практические работы, тесты для контроля знаний (АК-1 – АК-4, АК-6, АК-10 АК-12,СЛК-3,СЛК-5 ,СЛК-6)

Диагностика компетенций студента проводится в устной (ответы на занятиях, оценивание решения учебно-деловых ситуаций), письменной (контрольный опросы, письменное представление выполненных практических заданий, доклады и рефераты) и устно-письменной (зачет) формах. (АК-1 – АК-4, АК-6, АК-10 АК-12,СЛК-3,СЛК-5 ,СЛК-6)

Итоговая диагностика компетенций студента проводится с использованием контрольных вопросов, заданий и тестов, а также зачета (АК-1 – АК-4, АК-6, АК-10 АК-12,СЛК-3,СЛК-5 ,СЛК-6, ПК-1, ПК-2, ПК-4, ПК-6, ПК-7, ПК-11, ПК-15).

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Основы алгоритмизации и программирования	ИТ		протокол №3 от 15.10.2015
Основы теории систем	ПЭ	Нема 	протокол №3 от 15.10.2015
Теория передачи информации	ПЭ	Нема 	протокол №3 от 15.10.2015
Основы защиты информации»	ПЭ	Нема 	протокол №3 от 15.10.2015