

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Таныгин Максим Олегович

Должность: и.о. декана факультета фундаментальной и прикладной информатики

Дата подписания: 06.10.2022 11:17:42

Уникальный программный ключ:

65ab2aa0d384efe8480e6a4c688eddbc475e411a

## Аннотация к рабочей программе

### Дисциплины «Технологии и методы программирования»

### Направление подготовки бакалавров «10.03.01 - Информационная безопасность»

#### Цель преподавания дисциплины

является изучение основных этапов проектирования, кодирования, тестирования, отладки и оформления документации программного обеспечения с использованием современных средств программной инженерии.

#### Задачи изучения дисциплины

- разработка приложений баз данных на языках программирования Delphi с использованием BDE, ADO, технологий;
- ознакомление с основными моделями жизненного цикла программного обеспечения;
- ознакомление со стандартами, предназначенными для оценки качества процессов создания программного обеспечения;
- изучение методов нисходящей и восходящей разработки программного обеспечения;
- изучение методов повышения эффективности и технологичности программ;
- изучение эксплуатационных требований к программным продуктам, разделов документа «Техническое задание»;
- изучение методики разработки структуры программного обеспечения при структурном подходе, рассмотрение примеров разработки алгоритмов;
- изучение методики разработки структуры программного обеспечения при объектном подходе, рассмотрение примеров разработки диаграмм пакетов, диаграмм классов пакетов;
- получение практических навыков программирования на языке C++ с помощью структурного и объектного подходов;
- изучение классификации и функциональных возможностей CASE – средств,
- изучение типов пользовательских интерфейсов и этапов их разработки;
- изучение методов тестирования программного обеспечения;
- изучение методов отладки программного обеспечения;
- изучение видов программных документов.

#### Компетенции, формируемые в результате освоения дисциплины

**ПК-2** - Способность применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач.

## **Разделы дисциплины**

Разработка приложений баз данных на языке программирования Delphi.

Основные понятия технологии программирования.

Приемы обеспечения технологичности программных продуктов.

Определение требований к программному обеспечению.

Проектирование программного обеспечения при структурном подходе.

Структурное программирование на языке C++.

Проектирование программного обеспечения при объектном подходе.

Объектно-ориентированное программирование на языке C++.

Case – технологии.

Разработка пользовательских интерфейсов.

Тестирование программных продуктов.

Отладка программного обеспечения.

Составление программной документации.

МИНОБРНАУКИ РОССИИ

Юго-Западный государственный университет

УТВЕРЖДАЮ:  
Декан факультета  
**Фундаментальной и прикладной  
информатики**  
*(наименование ф-та полностью)*

 **Т. А. Ширабакина**  
*(подпись, инициалы, фамилия)*

« 2 » 02 2017 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

Технологии и методы программирования

*(наименование дисциплины)*

направление подготовки (специальность)

10.03.01

*(шифр согласно ФГОС)*

Информационная безопасность

*и наименование направления подготовки (специальности)*

Безопасность автоматизированных систем

*наименование профиля, специализации или магистерской программы*

форма обучения очная

*(очная, очно-заочная, заочная)*

Курск – 2017

Рабочая программа составлена в соответствии с Федеральным государственным образовательным стандартом высшего образования направления подготовки 10.03.01 «Информационная безопасность» и на основании учебного плана направления подготовки 10.03.01 «Информационная безопасность», одобренного Ученым советом университета протокол №5 от «20» 01 2017г.

Рабочая программа обсуждена и рекомендована к применению в образовательном процессе для обучения бакалавров по направлению подготовки 10.03.01 «Информационная безопасность» на заседании кафедры информационной безопасности, протокол № 9 « 1» 02 2017г.

*(наименование кафедры, дата, номер протокола)*

Зав. кафедрой \_\_\_\_\_ М. О. Таныгин

Разработчик программы  
к.т.н., доцент \_\_\_\_\_ К. А. Тезик

*(ученая степень и учебное звание, Ф.И.О.)*

Директор научной библиотеки \_\_\_\_\_ В.Г. Макаровская

Рабочая программа пересмотрена, обсуждена и рекомендована к применению в образовательном процессе на основании учебного плана направления подготовки 10.03.01 «Информационная безопасность», одобренного Ученым советом университета протокол №1 «28» 02 2017г. на заседании кафедры \_\_\_\_\_

*(наименование кафедры, дата, номер протокола)*

Зав. кафедрой \_\_\_\_\_

Рабочая программа пересмотрена, обсуждена и рекомендована к применению в образовательном процессе на основании учебного плана направления подготовки 10.03.01 «Информационная безопасность», одобренного Ученым советом университета протокол №5 «30» 01 2017г. на заседании кафедры ИБ, 29.06.2018, №12

*(наименование кафедры, дата, номер протокола)*

Зав. кафедрой \_\_\_\_\_

Рабочая программа пересмотрена, обсуждена и рекомендована к применению в образовательном процессе на основании учебного плана направления подготовки 10.03.01 «Информационная безопасность», одобренного Ученым советом университета протокол №1 «21» 01 2018г. на заседании кафедры ИБ, 27.06.2018, №11

*(наименование кафедры, дата, номер протокола)*

Зав. кафедрой \_\_\_\_\_

Программа пересмотрена, обсуждена и рекомендована к применению в образовательном процессе на основании учебного плана направления подготовки 10.03.01 – «Информационная безопасность», одобренного Ученым советом университета протокол № 7 «25» 02 2020 г. на заседании кафедры информационной безопасности. Протокол № 1 от «31» 08 2020 г.

Зав. кафедрой \_\_\_\_\_



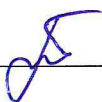
Программа пересмотрена, обсуждена и рекомендована к применению в образовательном процессе на основании учебного плана направления подготовки 10.03.01 – «Информационная безопасность», одобренного Ученым советом университета протокол № 7 «25» 02 2020 г. на заседании кафедры информационной безопасности. Протокол № 11 от «28» 06 2021 г.

Зав. кафедрой \_\_\_\_\_



Программа пересмотрена, обсуждена и рекомендована к применению в образовательном процессе на основании учебного плана направления подготовки 10.03.01 – «Информационная безопасность», одобренного Ученым советом университета протокол № 7 «25» 02 2020 г. на заседании кафедры информационной безопасности. Протокол № 11 от «30» 06 2022 г.

Зав. кафедрой \_\_\_\_\_



Программа пересмотрена, обсуждена и рекомендована к применению в образовательном процессе на основании учебного плана направления подготовки 10.03.01 – «Информационная безопасность», одобренного Ученым советом университета протокол №     «   »     20    г. на заседании кафедры информационной безопасности. Протокол №     от «   »     20    г.

Зав. кафедрой \_\_\_\_\_

Программа пересмотрена, обсуждена и рекомендована к применению в образовательном процессе на основании учебного плана направления подготовки 10.03.01 – «Информационная безопасность», одобренного Ученым советом университета протокол №     «   »     20    г. на заседании кафедры информационной безопасности. Протокол №     от «   »     20    г.

Зав. кафедрой \_\_\_\_\_

# **1. Цель и задачи дисциплины. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения образовательной программы**

## **1.1 Цель преподавания дисциплины**

Целью преподавания дисциплины «Технологии и методы программирования» является изучение основных этапов проектирования, кодирования, тестирования, отладки и оформления документации программного обеспечения с использованием современных средств программной инженерии.

## **1.2 Задачи дисциплины**

Основными обобщенными задачами дисциплины являются:

- разработка приложений баз данных на языке программирования Delphi с использованием BDE, ADO технологий;
- ознакомление с основными моделями жизненного цикла программного обеспечения;
- ознакомление со стандартами, предназначенными для оценки качества процессов создания программного обеспечения;
- изучение методов нисходящей и восходящей разработки программного обеспечения;
- изучение методов повышения эффективности и технологичности программ;
- изучение эксплуатационных требований к программным продуктам, разделов документа «Техническое задание»;
- изучение методики разработки структуры программного обеспечения при структурном подходе, рассмотрение примеров разработки алгоритмов;
- изучение методики разработки структуры программного обеспечения при объектном подходе, рассмотрение примеров разработки диаграмм пакетов, диаграмм классов пакетов;
- получение практических навыков программирования на языке C++ с помощью структурного и объектного подходов;
- изучение классификации и функциональных возможностей CASE – средств;
- изучение типов пользовательских интерфейсов и этапов их разработки;
- изучение методов тестирования программного обеспечения;
- изучение методов отладки программного обеспечения;
- изучение видов программных документов.

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

**Обучающиеся должны знать:**

- основные этапы развития технологии программирования;
- этапы разработки программного обеспечения;
- модели жизненного цикла программного обеспечения;
- основные спецификации программного обеспечения при структурном и объектном подходах;
- структуры алгоритмов;
- классификацию программного обеспечения, функциональные характеристики, возможности современных CASE – средств;
- эксплуатационные требования к программным продуктам;
- методы повышения эффективности и технологичности программ;
- основные виды пользовательских интерфейсов;
- стандарты, предназначенные для контроля качества процессов разработки программного обеспечения;
- методы тестирования программного обеспечения;
- методы отладки программного обеспечения;
- виды программных документов;
- модели данных, основные теоретические положения построения реляционных баз данных

**уметь:**

- разрабатывать техническое задание на проектирование программного обеспечения;
- принимать обоснованные решения по выбору архитектуры программного обеспечения, среды программирования, стандартов разработки;
- разрабатывать модели программного обеспечения, используемые на этапе определения спецификаций;
- проектировать программное обеспечение при структурном и объектном подходах;
- разрабатывать пользовательские интерфейсы;
- выполнять тестирование программного обеспечения;
- выполнять отладку программного обеспечения;
- разрабатывать приложения баз данных на языке программирования Delphi с использованием BDE , ADO технологий;
- оформлять программную документацию.

**владеть:**

- высокой мотивацией к выполнению профессиональной деятельности;
- навыками проектирования программного обеспечения при структурном и объектно-ориентированном подходе;
- навыками кодирования программного обеспечения на языке C++;

- методами тестирования программного обеспечения;
- методами и средствами отладки программного обеспечения;
- навыками разработки интерфейсов в интегрированных средах разработки Delphi и C++Builder.

В процессе изучения дисциплины «Технологии и методы программирования» происходит формирование следующих профессиональных компетенций:

– способность применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач (ПК-2).

## **2. Указание места дисциплины в структуре образовательной программы**

«Технологии и методы программирования» представляет дисциплину с индексом Б1.Б.22 базовой части учебного плана направления подготовки 10.03.01 Информационная безопасность, изучаемую на 2 курсе в 3 семестре.

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

Общая трудоемкость (объем) дисциплины составляет 5 зачетных единиц (з.е.), 180 академических часов.



Таблица 3.1 –Объём дисциплины

Виды учебной работы	Всего, часов
Общая трудоемкость дисциплины	180
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	56,65
в том числе:	
лекции	36
лабораторные занятия	18
практические занятия	0
экзамен	1,15
зачет	не предусмотрено
курсовая работа (проект)	1,5
расчетно-графическая (контрольная) работа	не предусмотрено
Аудиторная работа (всего):	54
в том числе:	
лекции	36
лабораторные занятия	18
практические занятия	0
Самостоятельная работа обучающихся (всего)	96,35
Контроль/экзамен (подготовка к экзамену)	27

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

##### **4.1 Содержание дисциплины**

Таблица 4.1 - Содержание дисциплины, структурированное по темам (разделам)

№	Раздел, (тема)	Содержание
---	----------------	------------

п/п	ДИСЦИПЛИНЫ	
1	2	3
1	Основные понятия технологии программирования	Основные этапы развития технологии программирования. Проблемы разработки сложных программных систем. Блочный – 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.
2	Приемы обеспечения технологичности программных продуктов	Понятие технологичности программного обеспечения. Модули и их свойства. Нисходящая и восходящая разработка программного обеспечения. Структурное программирование. Средства описания структурных алгоритмов. Стилль оформления программы. Эффективность и технологичность программ. Программирование с «защитой от ошибок». Сквозной структурный контроль.
3	Определение требований к программному обеспечению	Классификация программных продуктов по функциональному признаку. Основные эксплуатационные требования к программным продуктам. Предпроектные исследования предметной области. Разработка технического задания. Принципиальные решения начальных этапов проектирования.
4	Анализ требований и определение спецификаций программного обеспечения при структурном подходе	Спецификации программного обеспечения при структурном подходе: функциональные диаграммы, диаграммы потоков данных, диаграммы отношений компонентов данных.
5	Проектирование программного обеспечения при структурном подходе	Разработка структурной и функциональной схем. Использование метода пошаговой детализации для проектирования структуры программного обеспечения. Проектирование структур данных.
1	2	3
6	Структурное программирование на языке C++	Программирование алгоритмов линейной структуры. Программирование алгоритмов разветвленной структуры. Программирование алгоритмов циклической структуры. Указатели и ссылки. Статические и динамические массивы. Строки. Объявление и использование функций. Механизмы передачи аргументов. Аргументы функции main (). Рекурсия. Чтение и запись текстовых файлов. Неформатированный ввод-вывод данных в двоичном режиме. Считывание и запись в файл блоков данных. Обнаружение конца файла. Структуры и массивы структур.

7	Анализ требований и определение спецификаций программного обеспечения при объектном подходе	Спецификации программного обеспечения при объектном подходе: диаграммы вариантов использования, диаграммы последовательностей, диаграммы деятельности.
8	Проектирование программного обеспечения при объектном подходе	Разработка структуры программного обеспечения при объектном подходе. Диаграммы пакетов. Диаграммы классов пакетов. Компоновка программных компонентов. Проектирование размещения программных компонентов для распределенных программных систем.
9	Объектно-ориентированное программирование на языке C++	Объявление класса. Открытые и закрытые члены класса. Статические члены класса. Перегрузка методов. Конструкторы и деструкторы. Наследование классов и типы наследования. Многоуровневое и многократное наследование. Переопределение методов и виртуальные функции. Чисто виртуальные методы и
10	Case-технологии	Общая характеристика Case –средств. Классификация Case – средств. Определение потребностей в Case – средствах. Оценка и выбор Case –средств. Выполнение пилотного проекта. Практическое внедрение Case-средств. Обзор Case-средств: Silverrun, Oracle Designer, ERwin, BPwin, Rational Rose.
11	Разработка пользовательских интерфейсов	Типы пользовательских интерфейсов и этапы их разработки. Пользовательская и программная модели интерфейса. Классификация 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.
1	2	3
12	Тестирование программных продуктов	Виды контроля качества разрабатываемого программного обеспечения. Ручной контроль программного обеспечения. Структурное тестирование. Функциональное тестирование. Тестирование модулей и комплексное тестирование
13	Отладка программного обеспечения	Классификация ошибок. Методы отладки программного обеспечения. Методы и средства получения дополнительной информации. Основные этапы отладки программного обеспечения
14	Составление программной документации	Виды программных документов. Пояснительная записка. Руководство пользователя. Руководство системного программиста.

Таблица 4.2 - Содержание дисциплины и ее методическое обеспечение

№ п/п	Раздел учебной дисциплины	Виды учебной деятельности (в часах)			Учебно- методическ ие материалы	Формы текущего контроля успеваемости (по неделям семестра)	Компетен- ции	
		Лек (ча- сы)	№ лаб	№ пр.				
1	2	3	4	5	6	7	8	
1	Основные понятия технологии программирования	2	-	-	У-1, У-4, У-7, У-9, У-13	У-3, У-5, У-8, У-11,	С, Т1	ПК-2; ОК-5
2	Приемы обеспечения технологичности программных	2	-	-	У-1, У-4, У-7, У-9, У-13	У-3, У-5, У-8, У-11,	С, Т2	ПК-2
3	Определение требований к программному обеспечению	2	-	-	У-1, У-4, У-7, У-9, У-13	У-3, У-5, У-8, У-11,	С, Т3	ПК-2; ОК-5

1	2	3	4	5	6	7	8	
4	Анализ требований и определение спецификаций программного обеспечения при структурном	2	-	-	У-1, У-5, У-7, У-8, У-10, У-11, У-13		С, Т4	ПК-2
5	Проектирование программного обеспечения при структурном подходе	2	-	-	У-1, У-5, У-7, У-8, У-10, У-11, У-13		С, Т5	ПК-2
6	Структурное программирование на языке С++	2	1, 2, 3, 4, 5	-	У-3, У-12, МУ-2	У-6, У-14,	С, Т6	ПК-2; ОК-5

7	Анализ требований и определение спецификаций программного обеспечения при объектном подходе	2	-	-	У-1, У-2, У-5, У-8, У-10, У-11, У-12	С, Т7	ПК-2
8	Проектирование программного обеспечения при объектном подходе	2	-	-	У-1, У-2, У-5, У-8, У-10, У-11, У-12	С, Т8	ПК-2; ОК-5
9	Объектно-ориентированное программирование на языке С++	2	6, 7, 8	-	У-2, У-3, У-6, У-12, У-14, МУ-2	С, Т9	ПК-2
10	Case-технологии	4	-	-	У-1, У-8, У-11, У-13	С, Т11	ПК-2
11	Разработка пользовательских интерфейсов	6	9	-	У-1, У-6, У-14, У-15, У-16.	С, Т14	ПК-2; ОК-5
12	Тестирование программных продуктов	4	-	-	У-1, У-4, У-7, У-8, У-9, У-10, У-13	С, Т16	ПК-2; ОК-5
13	Отладка программного обеспечения	2	-	-	У-1, У-5, У-6, У-14	С, Т17	ПК-2

1	2	3	4	5	6	7	8
14	Составление программной документации	2	-	-	У-1, У-4, У-9, У-10	С, Т18	ПК-2
	Итого	36	18	-			

С – собеседование, Т - тест

## 4.2 Лабораторные работы и (или) практические занятия

### 4.2.1 Лабораторные работы

Таблица 4.3 - Лабораторные работы

№	Наименование лабораторной работы	Объем, час.
---	----------------------------------	-------------

1	Программирование алгоритмов базовых структур на языке программирования C++	2
2	Массивы в языке программирования C++	2
3	Функции в языке программирования C++	2
4	Файлы в языке программирования C++	2
5	Структуры в языке программирования C++	2
6	Объявление и применение классов в языке программирования C++. Статические методы класса.	2
7	Полиморфизм и наследование классов в языке программирования C++	2
8	Виртуальные функции и абстрактные классы в языке программирования C++	2
9	Интерфейсы со свободной навигацией в среде C++ Builder	2
Итого		18

### 4.3 Самостоятельная работа студентов (СРС)

Таблица 4.4 - Самостоятельная работа студентов

№ раздела (темы)	Наименование раздела (темы) дисциплины	Срок выполнения	Время, затрачиваемое на выполнение СРС, час.
1	Основные понятия технологии программирования	1 неделя	5
2	Приемы обеспечения технологичности программных продуктов	2 неделя	5
3	Определение требований к программному обеспечению	3 неделя	5
4	Анализ требований и определение спецификаций программного обеспечения при структурном подходе	4 неделя	5
5	Проектирование программного обеспечения при структурном подходе	5 неделя	5
6	Структурное программирование на языке C++	6 неделя	5
7	Анализ требований и определение спецификаций программного обеспечения при объектном подходе	7 неделя	5
8	Проектирование программного обеспечения при объектном подходе	8 неделя	5

9	Объектно-ориентированное программирование на языке C++	9 неделя	5
10	Case-технологии	11 неделя	5
11	Разработка пользовательских интерфейсов	14 неделя	5
12	Тестирование программных продуктов	16 неделя	5
13	Отладка программного обеспечения	17 неделя	5
14	Составление программной документации	18 неделя	5
15	Выполнение и защита курсового проекта	1-18 недели	26,35
Итого			96,35

## 5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Студенты могут при самостоятельном изучении отдельных тем и вопросов дисциплин пользоваться учебно-наглядными пособиями, учебным оборудованием и методическими разработками кафедры в рабочее время, установленное «Правилами внутреннего распорядка работников».

Учебно-методическое обеспечение для самостоятельной работы обучающихся по данной дисциплине организуется:

библиотекой университета:

- библиотечный фонд укомплектован учебной, методической, научной, периодической, справочной и художественной литературой в соответствии с данной РПД;

- имеется доступ к основным информационным образовательным ресурсам, информационной базе данных, в том числе библиографической, возможность выхода в Интернет.

кафедрой:

- путем обеспечения доступности всего необходимого учебно-методического и справочного материала за счёт выкладывания на сайт кафедры ИБ в интернете (адрес [http://www.swsu.ru/structura/up/fivt/k\\_tele/index.php](http://www.swsu.ru/structura/up/fivt/k_tele/index.php));

- путем предоставления сведений о наличии учебно-методической литературы, современных программных средств;

путем разработки :

- методических рекомендаций, пособий по организации

самостоятельной работы студентов;

– заданий для самостоятельной работы;

– тем рефератов и докладов;

– вопросов и задач к экзамену;

– методических указаний к выполнению лабораторных работ и т.д.

*типографией университета:*

– помощь авторам в подготовке и издании научной, учебной и методической литературы;

– удовлетворение потребности в тиражировании научной, учебной и методической литературы.

## 6. Образовательные технологии

В соответствии с требованиями ФГОС и Приказа Министерства образования и науки РФ от 05 апреля 2017 г. № 301 по направлению подготовки 10.03.01 «Информационная безопасность», реализация компетентностного подхода должна предусматривать широкое использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков студентов. Удельный вес занятий, проводимых в интерактивных формах, составляет 11 % от аудиторных занятий согласно УП.

Таблица 6.1 - Интерактивные образовательные технологии, используемые при проведении аудиторных занятий

№	Наименование раздела (темы лекции, практического или лабораторного занятия)	Используемые интерактивные образовательные технологии	Объем, час
1	2	3	4
1	<b>Лабораторная работа №1.</b> Программирование алгоритмов базовых структур на языке программирования С++	Анализ конкретных ситуаций	1
2	<b>Лабораторная работа №2.</b> Массивы в языке программирования С++	Анализ конкретных ситуаций	1
3	<b>Лабораторная работа № 3.</b> Функции в языке программирования	Анализ конкретных ситуаций	1
4	<b>Лабораторная работа № 4.</b> Файлы в языке программирования С++	Анализ конкретных ситуаций	1
5	<b>Лабораторная работа №5.</b> Структуры в языке	Анализ конкретных ситуаций	1



6	<b>Лабораторная работа №6.</b> Объявление и применение классов в языке программирования C++. Статические методы класса.	Анализ конкретных ситуаций	1
Итого			6

## 7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

### 7.1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Таблица 7.1 – Этапы формирования компетенций

Код и содержание компетенции	Этапы* формирования компетенций и дисциплины (модули), при изучении которых формируется данная компетенция		
	начальный	основной	завершающий
Способность применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач (ПК-2).	Языки программирования;  Технологии и методы программирования;  Информационные технологии;  Ознакомительная практика	Введение в криптографию;  Криптографические методы защиты информации;  Методы защиты программного обеспечения;  Основы риверсинжиниринга программных средств	Эксплуатационная практика;  Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты

<p>способность понимать социальную значимость своей будущей профессии, обладать высокой мотивацией к выполнению профессиональной деятельности в области обеспечения информационной безопасности и защиты интересов личности, общества и государства, соблюдать нормы профессиональной этики. (ОК-5)</p>	<p>Основы информационной безопасности;</p> <p>Языки программирования;</p> <p>Технологии и методы программирования;</p> <p>Введение в направление подготовки и планирование профессиональной карьеры;</p> <p>История информационного противоборства;</p> <p>Практика по получению первичных профессиональных навыков и компетенций</p>	Социология	<p>Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты</p>
---	---	------------	---

\*Этапы для РПД всех форм обучения определяются по учебному плану очной формы обучения следующим образом:

Этап	Учебный план очной формы обучения/ семестр изучения дисциплины		
	Бакалавриат	Специалитет	Магистратура
<i>Начальный</i>	1-3 семестры	1-3 семестры	1 семестр
<i>Основной</i>	4-6 семестры	4-6 семестры	2 семестр
<i>Завершающий</i>	7-8 семестры	7-10 семестры	3-4 семестр

\*\* Если при заполнении таблицы обнаруживается, что один или два этапа не обеспечены дисциплинами, практиками, НИР, необходимо:

- при наличии дисциплин, изучающихся в разных семестрах, – распределить их по этапам в зависимости от № семестра изучения (начальный этап соответствует более раннему семестру, основной и завершающий – более поздним семестрам);

- при наличии дисциплин, изучающихся в одном семестре, – все дисциплины указать для всех этапов.

## **7.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

Таблица 7.2 – Показатели, критерии и шкала оценивания компетенций

Код компетенции/ этап (указывается название этапа из п.7.1)	Показатели оценивания компетенций	Критерии и шкала оценивания компетенций		
		Пороговый уровень (удовлетворительно)	Продвинутый уровень (хорошо)	Высокий уровень (отлично)
1	2	3	4	5

<p>Способность применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач.</p> <p>ПК-2/ начальный</p>	<p>1. Доля освоенных обучающимся знаний, умений, навыков от общего объема ЗУН, установленных в п. 1.3 РПД</p> <p>2. Качество освоенных обучающимся знаний, умений, навыков</p> <p>3. Умение применять знания, умения, навыки в типовых и нестандартных ситуациях</p>	<p><b>Знать:</b> способы обеспечения технологичности и повышения эффективности ПО, методику проектирования ПО</p> <p><b>Уметь:</b> разрабатывать структурные схемы алгоритмов, структурные и функциональные схемы программных систем</p> <p><b>Владеть навыками:</b> разработки спецификаций, проектирования ПО</p>	<p><b>Знать:</b> способы обеспечения технологичности и повышения эффективности ПО, методику проектирования, кодирования, отладки ПО</p> <p><b>Уметь:</b> разрабатывать структурные схемы алгоритмов, структурные и функциональные схемы программных систем, программы на языках C++, Delphi</p> <p><b>Владеть навыками:</b> разработки спецификаций, проектирования, разработки и сопровождения ПО с использованием современных средств программной инженерии</p>	<p><b>Знать:</b> способы обеспечения технологичности и повышения эффективности ПО, методику проектирования, кодирования, отладки ПО, а также методические подходы к разработке тестов для контроля ПО</p> <p><b>Уметь:</b> разрабатывать структурные схемы алгоритмов, структурные и функциональные схемы программных систем, программы на языках C++, Delphi, а также разрабатывать тесты для контроля ПО</p> <p><b>Владеть навыками:</b> разработки спецификаций, проектирования, разработки и сопровождения ПО с использованием современных средств программной инженерии, разработки тестов для контроля ПО</p>
---	--	---	---	---

<p>Способность понимать социальную значимость своей будущей профессии, обладать высокой мотивацией к выполнению профессиональной деятельности и в области обеспечения информационной безопасности и защиты интересов личности, общества и государства, соблюдать нормы профессиональной этики.</p> <p>ОК-5/ начальный</p>	<p>1. Доля освоенных обучающимся знаний, умений, навыков от общего объема ЗУН, установленных в п. 1.3 РПД</p> <p>2. Качество освоенных обучающимся знаний, умений, навыков</p> <p>3. Умение применять знания, умения, навыки в типовых и нестандартных ситуациях</p>	<p><b>Знать:</b> основные принципы системы информационной безопасности, основные направления деятельности.</p> <p><b>Уметь:</b> выработать управленческие решения в сфере профессиональной деятельности</p> <p><b>Владеть навыками:</b> способность к работе в коллективе и кооперации с коллегами</p>	<p><b>Знать:</b> основные принципы системы информационной безопасности, основные направления деятельности.</p> <p><b>Уметь:</b> в качестве руководителя подразделения, формировать цели команды, принимать организационно-управленческие решения в ситуациях риска и нести за них ответственность</p> <p><b>Владеть навыками:</b> к освоению новых образцов программных, технических средств и информационных технологий</p>	<p><b>Знать:</b> основные принципы системы информационной безопасности; как восстановить работоспособность системы защиты информации при возникновении нештатных ситуаций</p> <p><b>Уметь:</b> организовывать работу малых коллективов исполнителей, выработать и реализовывать управленческие решения в сфере профессиональной деятельности</p> <p><b>Владеть навыками:</b> защиты и восстановления работоспособности, подсистем информационной безопасности автоматизированной системы</p>
---	--	--	--	--

**7.3 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**

Таблица 7.3 Паспорт комплекта оценочных средств для текущего контроля

№ п/п	Раздел (тема) дисциплины	Код контролируемой компетенции (или её части)	Технология формирования	Оценочные средства		Описание шкал оценивания
				наименование	№№ заданий	
1	2	3	4	5	6	7
1	Основные понятия технологии программирования	ПК-2; ОК-5	Лекция, СРС	Собеседование	1-6	Согласно таблице 7.2
				Тестирование	1-4	
2	Приемы обеспечения технологичности программных	ПК-2	Лекция, СРС	Собеседование	7-12	Согласно таблице 7.2
				Тестирование	5-8	
3	Определение требований к программному обеспечению	ПК-2; ОК-5	Лекция, СРС	Собеседование	13-16	Согласно таблице 7.2
				Тестирование	9-12	
4	Анализ требований и определение спецификаций программного обеспечения при структурном	ПК-2	Лекция, СРС	Собеседование	17	Согласно таблице 7.2
				Тестирование	13-14	
5	Проектирование программного	ПК-2	Лекция, СРС	Собеседование	18	Согласно таблице 7.2

	обеспечения при структурном			Тестирование	15-16	
6	Структурное программирование на языке C++	ПК-2; ОК-5	Лекция, лаб. №1, лаб. №2, лаб. №3, лаб. №4, лаб. №5, СРС	Собеседование	19-32	Согласно таблице 7.2
				Тестирование	17-20	
				Контрольные вопросы к лаб. №1	1-14	
				Контрольные вопросы к лаб. №2	15-22, 30-33	
				Контрольные вопросы к лаб. №3	23-29	
				Контрольные вопросы к лаб. №4	34-37	
				Контрольные вопросы к лаб. №5	38-41	

7	Анализ требований и определение спецификаций программного обеспечения при объектном подходе	ПК-2	Лекция, СРС	Собеседование	33	Согласно таблице 7.2
				Тестирование	21-22	
8	Проектирование программного обеспечения при объектном подходе	ПК-2; ОК-5	Лекция, СРС	Собеседование	34	Согласно таблице 7.2
				Тестирование	23-24	
9	Объектно-ориентированное программирование на языке	ПК-2	Лекция, лаб. №6, лаб. №7, лаб. №8, СРС	Собеседование	35-41	Согласно таблице 7.2
				Тестирование	25-28	
				Контрольные вопросы к лаб.	42-46, 48-51	

	C++			Контрольные вопросы к лаб.	47, 52-53, 55-56	
				Контрольные вопросы к лаб.	54,57-58	
10	Case-технологии	ПК-2	Лекция, СРС	Собеседование	42-43	Согласно таблице 7.2
				Тестирование	29-32	
11	Разработка пользовательских интерфейсов	ПК-2; ОК-5	Лекция, лаб. №9, СРС	Собеседование	44-47	Согласно таблице 7.2
				Тестирование	33-36	
				Контрольные вопросы к лаб.	1-12	
12	Тестирование программных продуктов	ПК-2; ОК-5	Лекция, СРС	Собеседование	48-49	Согласно таблице 7.2
				Тестирование	37-40	
13	Отладка программного обеспечения	ПК-2	Лекция, СРС	Собеседование	50	Согласно таблице 7.2
				Тестирование	41-44	
14	Составление программной документации	ПК-2	Лекция, СРС	Собеседование	51	Согласно таблице 7.2
				Тестирование	45-48	

### Типовые задания для промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в форме экзамена в форме бланкового тестирования.

Умения, навыки и компетенции проверяются с помощью задач (ситуационных, производственных или кейсового характера) и различного вида конструкторов. Часть умений, навыков и компетенций прямо не отражена в формулировках задач, но они могут быть проявлены обучающимися при их решении.



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

Процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, регулируются следующими нормативными актами университета:

- Положение П 02.016–2015 «О балльно-рейтинговой системе оценки качества освоения образовательных программ»;
- Методические указания, используемые в образовательном процессе, указанные в списке литературы.

Для *текущего контроля* по дисциплине в рамках действующей в университете балльно-рейтинговой системы применяется следующий порядок начисления баллов .

Таблица 7.4 – Порядок начисления баллов в рамках БРС

Форма контроля	Минимальный балл		Максимальный балл	
	балл	примечание	балл	примечание
1	2	3	4	5
Выполнение лабораторной работы №1 «Программирование алгоритмов базовых структур на языке программирования	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»
Выполнение лабораторной работы №2 «Массивы в языке программирования C++ »	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»
Выполнение лабораторной работы №3 «Функции в языке программирования C++ »	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»
Выполнение лабораторной работы №4 «Файлы в языке программирования C++ »	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»

Выполнение лабораторной работы №5 «Структуры в языке программирования С++»	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»
Выполнение лабораторной работы №6 «Объявление и применение классов в языке программирования С++. Статические	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»
Выполнение лабораторной работы №7 «Полиморфизм и наследование классов в языке программирования	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»

1	2	3	4	5
Выполнение лабораторной работы №8 «Виртуальные функции и абстрактные классы в языке	3	Выполнил, но «не защитил»	4	Выполнил и «защитил»
Выполнение лабораторной работы №9 «Интерфейсы со свободной навигацией в среде	3	Выполнил, но «не защитил»	4	Выполнил и «защитил»
СРС	4		12	
Итого	24		48	
Посещаемость	0		16	
Экзамен	0		36	
Итого	24		100	

При итоговом контроле в форме бланкового тестирования студенту предлагается 15 вопросов по различным темам курса. Каждый вопрос оценивается в 4 условных балла. Полученную итоговую сумму условных баллов (максимум 60) переводят в баллы на экзамене (максимум 36) путём умножения на 0.6 и округления до целого значения. Пример билета в тестовой форме приведён в приложении Д..

### **Критерии оценки курсового проекта**

1. Формальные критерии (0-30 баллов):
  - оформление титульного листа, технического задания, текста, приложений.
  - оформление списка литературы;
  - грамматика, пунктуация;
  - соблюдение графика подготовки и сроков сдачи работы.
2. Содержательные критерии (0-50 баллов):
  - соответствие работы заданию;
  - структура работы, сбалансированность разделов;
  - использование литературы;
  - степень самостоятельности работы;
  - стиль изложения.
3. Защита (0-20 баллов):
  - раскрытие содержания работы;
  - оперирование профессиональной терминологией;
  - ответы на вопросы.

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

### **8.1 Основная учебная литература**

1. Иванова Г. С. Технология программирования [Текст]: учебник / Г. С. Иванова.- М.: КНОРУС, 2011. – 336 с.
2. Иванова Г.С. Объектно-ориентированное программирование: [Текст] : учебник / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев. - 3-е изд., стер. - М.: МГТУ им. Баумана, 2007. - 368 с.
3. Технология программирования [Электронный ресурс]: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, М.П. Беляев, Ю.В. Минин.- Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2013. - 173 с. // Режим доступа - <http://biblioclub.ru/index.php?page=book&id=277802>
4. Терехов А.Н. Технология программирования [Электронный ресурс]: учебное пособие / А.Н. Терехов. - 2-е изд. - М. : Интернет-Университет Информационных Технологий, 2007. - 149 с. // Режим доступа - <http://biblioclub.ru/index.php?page=book&id=233491>
5. Программирование и основы алгоритмизации [Электронный ресурс]: учебное пособие / В.К. Зольников, П.Р. Машевич, В.И. Анциферова, Н.Н. Литвинов. - Воронеж : Воронежская государственная лесотехническая академия, 2011. - 341 с. // Режим доступа - <http://biblioclub.ru/index.php?page=book&id=142309>

## 8.2 Дополнительная учебная литература

6. Павловская Т. А. С/С++. Программирование на языке высокого уровня [Текст] : учебник / Т. А. Павловская. - СПб. : Питер, 2006. - 461 с.
7. Лаврищева Е.М., Петрухин В.А. Методы и средства инженерии программного обеспечения [Электронный ресурс]: учебник. - М.: МФТИ (ГУ), 2006. - 304 с. / Электронная библиотека «Единое окно доступа к образовательным ресурсам» - <http://window.edu.ru>
8. Камаев В. А. Технология программирования [Текст] : учебник для студентов вузов / В. А. Камаев, В. В. Костерин. - М.: Высшая школа, 2005. - 359 с.
9. Благодатских В. А. Стандартизация разработки программных средств [Текст]: учебное пособие / В. А. Благодатских, В. А. Волнин, К. Ф. Посакалов; под ред. О. С. Разумова. – М.: Финансы и статистика, 2005.-288 с.
10. Липаев В. В. Методы обеспечения качества крупномасштабных программных средств[Текст]. – М.: СИНТЕГ, 2003. -520 с.
11. Вендров А. М. Проектирование программного обеспечения экономических информационных систем [Текст]: учебник для студ. вуз. / А. М. Вендров. - М. : Финансы и статистика, 2000. - 352 с.
12. Буч Г. Объектно-ориентированный анализ и проектирование с примерами и приложениями на С++ = Object-oriented analysis and design with applications second edition [Текст] / Г. Буч ; Пер. с англ. - 2-е изд. - СПб. : Бинум, 2001 ; СПб. : Невский диалект. - 560 с.
13. Орлов С. А. Технологии разработки программного обеспечения. Разработка сложных программных систем [Текст]: учебник / С. А. Орлов. - 3-е изд. - СПб. : Питер, 2004. - 527 с.
14. Зотов И. В. Разработка программ на С++. Теория, примеры решения задач, эффективность [Текст] : учебное пособие / Т. А. Ширабакина. - Курск : Курск. гос. техн. ун-т, 2003. - 172 с.
15. Ачкасов В.Ю. Программирование баз данных в Delphi [Электронный ресурс] : курс / В.Ю. Ачкасов –М: Национальный Открытый Университет "ИНТУИТ", 2010. - 382 с. // Режим доступа - <http://biblioclub.ru/index.php?page=book&id=233558>
16. Хомоненко А. Д. Самоучитель Delphi. – 2-е изд. перераб. и доп. [Текст] / А. Д. Хомоненко, В. Э. Гофман. – СПб.: БХВ-Петербург, 2008.- 576 с.

## 8.3 Перечень методических указаний

1. Разработка приложений для управления базами данных на языке Delphi [Электронный ресурс]: методические указания по выполнению курсового проекта по дисциплине «Технологии и методы программирования», для студентов направления подготовки бакалавров

090900.62 / Юго-Зап. гос. ун-т; сост. К. А. Тезик. - Электрон. текстовые дан. - Курск: ЮЗГУ, 2015. - 33 с.

2. Реализация структурного и объектного подходов на языке программирования С++ [Электронный ресурс] : методические указания по выполнению лабораторных работ по дисциплине «Технологии и методы программирования» / Юго-Зап. гос. ун-т; сост.: К. А. Тезик. Курск: ЮЗГУ, 2014. - 63 с.

## 8.4 Другие учебно-методические материалы

### Периодические издания

1. Журнал «Программные продукты и системы»;
2. Журнал «ПРОграммист»;
3. Журнал «Программирование»;
4. Журнал «Вычислительные методы и программирование»;
5. Журнал «Программная инженерия».

## 9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. <http://e.lanbook.com> - Электронно-библиотечная система «Лань».
2. <http://www.iqlib.ru> - Электронно-библиотечная система IQLib.
3. <http://window.edu.ru> -Электронная библиотека «Единое окно доступа к образовательным ресурсам».
4. <http://biblioclub.ru> – Электронно-библиотечная система «Университетская библиотека онлайн».
5. <http://www.delphisources.ru> – веб-портал о программировании.
6. <https://code-live.ru> – веб-портал о программировании.

## 10. Методические указания для обучающихся по освоению дисциплины

Основными видами аудиторной работы студента при изучении дисциплины «Технологии и методы программирования» являются лекции и лабораторные занятия. Студент не имеет права пропускать занятия без уважительных причин.

На лекциях излагаются и разъясняются основные понятия темы, связанные с ней теоретические и практические проблемы, даются рекомендации для самостоятельной работы. В ходе лекции студент должен внимательно слушать и конспектировать материал.

Изучение наиболее важных тем или разделов дисциплины завершают лабораторные занятия, которые обеспечивают: контроль подготовленности

студента; закрепление учебного материала; приобретение опыта устных публичных выступлений, ведения дискуссии, в том числе аргументации и защиты выдвигаемых положений и тезисов.

Лабораторному занятию предшествует самостоятельная работа студента, связанная с освоением материала, полученного на лекциях, и материалов, изложенных в учебниках и учебных пособиях, а также литературе, рекомендованной преподавателем.

Качество учебной работы студентов преподаватель оценивает по результатам тестирования, собеседования, защиты отчетов по лабораторным и работам.

Преподаватель уже на первых занятиях объясняет студентам, какие формы обучения следует использовать при самостоятельном изучении дисциплины «Технологии и методы программирования»: конспектирование учебной литературы и лекции, составление словарей понятий и терминов и т. п.

В процессе обучения преподаватели используют активные формы работы со студентами: чтение лекций, привлечение студентов к творческому процессу на лекциях, промежуточный контроль путем отработки студентами пропущенных лекций, участие в групповых и индивидуальных консультациях (собеседованиях). Эти формы способствуют выработке у студентов умения работать с учебником и литературой. Изучение литературы и справочной документации составляет значительную часть самостоятельной работы студента. Это большой труд, требующий усилий и желания студента. В самом начале работы над книгой важно определить цель и направление этой работы. Прочитанное следует закрепить в памяти. Одним из приемов закрепления освоенного материала является конспектирование, без которого немислима серьезная работа над литературой. Систематическое конспектирование помогает научиться правильно, кратко и четко излагать своими словами прочитанный материал.

Самостоятельную работу следует начинать с первых занятий. От занятия к занятию нужно регулярно прочитывать конспект лекций, знакомиться с соответствующими разделами учебника, читать и конспектировать литературу по каждой теме дисциплины. Самостоятельная работа дает студентам возможность равномерно распределить нагрузку, способствует более глубокому и качественному усвоению учебного материала. В случае необходимости студенты обращаются за консультацией к преподавателю по вопросам дисциплины «Технологии и методы программирования» с целью усвоения и закрепления компетенций.

Основная цель самостоятельной работы студента при изучении дисциплины «Технологии и методы программирования» - закрепить теоретические знания, полученные в процессе лекционных занятий, а также сформировать практические навыки самостоятельного анализа особенностей дисциплины.

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

Интегрированная среда разработки программ Lazarus  
Microsoft Office 2016. Лицензионный договор №S0000000722 от 21.12.2015 г. с ООО «АйТи46», лицензионный договор №K0000000117 от 21.12.2015 г. с ООО «СМСКанал»,  
Kaspersky Endpoint Security Russian Edition, лицензия 156A-140624-192234,  
Windows 7, договор IT000012385

**12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Учебная аудитория для проведения занятий лекционного типа и лаборатории кафедры информационной безопасности, оснащенные учебной мебелью: столы, стулья для обучающихся; стол, стул для преподавателя; доска. Компьютеры (12 шт) CPU AMD-Phenom, ОЗУ 16 GB, HDD 2 Tb, монитор Aок 21". Проекционный экран на штативе; Мультимедиацентр: ноут-бук ASUS X50VLPMD-T2330/14"/1024Mb/ 160Gb/сумка/проектор inFocus IN24+.

**13. Лист дополнений и изменений, внесенных в рабочую программу дисциплины**

Номер изменения	Номера страниц				Всего страниц	Дата	Основание для изменения и подпись лица, проводившего изменения
	Изменённых	Заменённых	Аннулированных	новых			



## **Приложение А**

### **Вопросы для подготовки к экзамену**

- 1.** Основные этапы развития технологии программирования.  
Проблемы разработки сложных программных систем.
- 2.** Блочный – иерархический подход к созданию сложных систем.
- 3.** Жизненный цикл и этапы разработки программного обеспечения.
- 4.** Эволюция моделей жизненного цикла программного обеспечения.
- 5.** Ускорение разработки программного обеспечения. Технология RAD.
- 6.** Оценка качества процессов создания программного обеспечения.
- 7.** Понятие технологичности программного обеспечения.  
Модули и их свойства.
- 8.** Нисходящая и восходящая разработка программного обеспечения.
- 9.** Структурное программирование. Средства описания структурных алгоритмов. Псевдокоды.
- 10.** Стиль оформления программы. Правила именования объектов программы. Правила оформления модулей. Стиль оформления текстов модулей.
- 11.** Эффективность и технологичность программ. Способы экономии памяти. Способы уменьшения времени выполнения.
- 12.** Программирование с «защитой от ошибок». Сквозной структурный контроль.
- 13.** Классификация программных продуктов по функциональному признаку.
- 14.** Основные эксплуатационные требования к программным продуктам.
- 15.** Предпроектные исследования предметной области. Разработка технического задания.
- 16.** Принципиальные решения начальных этапов проектирования.
- 17.** Типы пользовательских интерфейсов и этапы их разработки.
- 18.** Классификация диалогов и общие принципы их разработки при проектировании интерфейсов. Графы абстрактного диалога.
- 19.** Основные компоненты графических пользовательских интерфейсов и реализация диалогов в них.
- 20.** Пользовательские интерфейсы прямого манипулирования и их проектирование.

21. Спецификации программного обеспечения при структурном подходе: функциональные диаграммы, диаграммы потоков данных, диаграммы отношений компонентов данных.
22. Спецификации программного обеспечения при объектном подходе: диаграммы вариантов использования, диаграммы последовательностей, диаграммы деятельности.
23. Разработка структурной и функциональной схем программного обеспечения. Метод пошаговой детализации.
24. Виды программных документов. Пояснительная записка. Руководство пользователя. Руководство системного программиста.
25. Разработка структуры программного обеспечения при объектном подходе. Диаграмма пакетов. Диаграмма классов.
26. Общая характеристика Case –средств. Классификация Case – средств.
27. Обзор Case-средств: Silverrun, Oracle Designer, ERwin, BPwin, Rational Rose.
28. Структурное тестирование. Покрытие операторов. Покрытие решений. Покрытие условий. Комбинаторное покрытие условий.
29. Функциональное тестирование. Эквивалентное разбиение. Анализ граничных значений. Анализ причинно-следственных связей. Предположение об ошибке.
30. Методы отладки программного обеспечения. Методы и средства получения дополнительной информации об ошибке.
31. Структура программы в языке программирования C++. Объявление и инициализация переменных. Базовые типы данных.
32. Арифметические операторы в языке программирования C++. Операторы сравнения. Логические операторы. Побитовые операторы.
33. Условный оператор if в языке программирования C++.
34. Условный оператор switch в языке программирования C++.
35. Оператор цикла for в языке программирования C++. Вложенные циклы.
36. Операторы цикла while и do - while в языке программирования C++.
37. Статические массивы в языке программирования C++. Инициализация статических массивов.
38. Строки в языке программирования C++. Функции для работы со строками.
39. Динамическое выделение памяти в языке программирования C++. Динамические массивы.
40. Файлы в языке программирования C++. Методы записи информации в файл и чтения информации из файла.

41. Объявление и использование функций в языке программирования C++. Понятие прототипа функции. Возвращение функцией значения.
42. Механизмы передачи аргументов по ссылке и по значению в языке программирования C++.
43. Аргументы функции `main()` : `argv` и `argc`.
44. Структуры в языке программирования C++. Массивы структур.
45. Классы и объекты в языке программирования C++. Объявление класса. Открытые и закрытые члены класса.

## Приложение Б.

### ПРИЛОЖЕНИЕ В Образец экзаменационного билета в тестовой форме.

ЮГО-ЗАПАДНЫЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ	
Факультет ФиПИ	Утверждено на заседании
Направление подготовки 10.03.01	кафедры ИБ,
курс 3, семестр 6	Протокол № __ от ____ 201__ г.
Дисциплина <b>«Безопасность операционных систем»</b>	Зав. кафедрой _____ М.О. Таныгин
<p>1. Какой язык программирования реализует только объектный подход к программированию?</p> <ol style="list-style-type: none"> <li>a. Java</li> <li>b. JavaScript</li> <li>c. C++</li> <li>d. Delphi</li> </ol>	
<p>2. Выберите правильную последовательность основных этапов разработки программного обеспечения.</p> <ol style="list-style-type: none"> <li>1. Постановка задачи, анализ требований и определение спецификаций, проектирование, реализация, сопровождение</li> <li>2. Постановка задачи, проектирование, анализ требований и определение спецификаций, реализация, сопровождение</li> <li>3. Постановка задачи, анализ требований и определение спецификаций, проектирование, сопровождение, реализация</li> <li>4. Последовательность этапов разработки программного обеспечения может быть любой.</li> </ol>	
<p>3. Какая модель разработки программного обеспечения предусматривает создание программного обеспечения не сразу, а итерационно с использованием метода, базирующегося на создании прототипа?</p> <ol style="list-style-type: none"> <li>1. Спиральная модель</li> <li>2. Каскадная модель</li> <li>3. Модель с промежуточным контролем</li> <li>4. Любая модель</li> </ol>	
<p>4. Наиболее высокий уровень организации – разработчика программного обеспечения – это:</p> <ol style="list-style-type: none"> <li>1) Оптимизирующий уровень</li> <li>2) Повторяемый уровень</li> <li>3) Определенный уровень</li> <li>4) Управляемый уровень</li> </ol>	

<p>5. Какие типы сцепления модулей являются допустимыми?</p> <ol style="list-style-type: none"><li>1. По данным, по образцу, по управлению</li><li>2. По данным, по образцу, по общей области</li><li>3. По управлению, по общей области, по содержимому</li><li>4. По образцу, по управлению, по общей области</li></ol>
<p>6. При промышленном изготовлении сложного программного обеспечения следует использовать:</p> <ol style="list-style-type: none"><li>1. Нисходящий подход.</li><li>2. Восходящий подход.</li><li>3. Это зависит от того, используется ли в программе управление базой данных.</li><li>4. Безразлично, какой подход выбрать: восходящий или нисходящий.</li></ol>
<p>7. Какой оператор в языке Паскаль не соответствует принципу структурного программирования?</p> <ol style="list-style-type: none"><li>1. Goto</li><li>2. If...then...else</li><li>3. For</li><li>4. While...do</li></ol>
<p>8. Какая рекомендация позволяет экономить память, а не время выполнения программы?</p> <ol style="list-style-type: none"><li>1. Размещать массивы, используемые ограниченное время, в динамической памяти и удалять их при завершении обработки.</li><li>2. Выносить вычисление константных, т. е. не зависящих от параметров цикла, выражений из циклов.</li><li>3. Избегать операций умножения и деления, заменяя их сложением, вычитанием и сдвигами.</li><li>4. Оптимизировать запись условных выражений – исключить лишние проверки.</li></ol>
<p>9. Какие программные продукты относят к системным?</p> <ol style="list-style-type: none"><li>1. Операционные системы, оболочки, утилиты.</li><li>2. Автоматизированные системы управления реального времени.</li><li>3. CASE – средства, среды разработки, отладочные средства.</li><li>4. Бухгалтерские системы, издательские системы, профессиональные графические системы.</li></ol>
<p>10. Документ, в котором сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки называется:</p> <ol style="list-style-type: none"><li>1. Техническим заданием</li><li>2. Пояснительной запиской</li><li>3. Руководством пользователя</li><li>4. Руководством системного программиста</li></ol>
<p>11. Языки низкого уровня типа Ассемблера</p> <ol style="list-style-type: none"><li>1. Обеспечивают оптимальные с точки зрения времени выполнения и объема</li></ol>

<p>памяти программы.</p> <ol style="list-style-type: none"><li>2. Поддерживают принцип структурного программирования.</li><li>3. Поддерживают объектный подход к программированию.</li><li>4. Используют для разработки сложных программных систем прикладного назначения.</li></ol>
<p>12. Какие модели этапа анализа и определения спецификаций относятся к структурному подходу?</p> <ol style="list-style-type: none"><li>1. Диаграммы потоков данных.</li><li>2. Диаграммы вариантов использования.</li><li>3. Диаграммы последовательностей.</li><li>4. Диаграммы деятельности.</li></ol>
<p>13. Выберите вариант, при котором отношение между сущностями относится к типу «один – ко- многим»</p> <ol style="list-style-type: none"><li>1. Куратор и студент.</li><li>2. Студент и номер зачетной книжки.</li><li>3. Преподаватель и студент.</li><li>4. Преподаватель и учебная дисциплина</li></ol>
<p>14. При проектировании программного обеспечения при структурном подходе какие схемы более информативны: структурные или функциональные?</p> <ol style="list-style-type: none"><li>1. Функциональные.</li><li>2. Структурные.</li><li>3. Это зависит от характера решаемой задачи.</li><li>4. Структурные и функциональные схемы – это одно и то же.</li></ol>
<p>15. Какие модели этапа анализа и определения спецификаций относятся к объектному подходу?</p> <ol style="list-style-type: none"><li>1. Диаграммы последовательностей.</li><li>2. Функциональные диаграммы.</li><li>3. Диаграммы потоков данных.</li><li>4. Диаграммы отношений компонентов данных.</li></ol>
<p>16. С какой диаграммы следует начинать проектирование программного обеспечения при объектном подходе?</p> <ol style="list-style-type: none"><li>1. Диаграмма пакетов.</li><li>2. Уточненная диаграмма классов.</li><li>3. Диаграмма последовательностей этапа проектирования.</li><li>4. Любая диаграмма.</li></ol>
<p>17. Какой член является общим для всех объектов класса?</p> <ol style="list-style-type: none"><li>1. Статический.</li><li>2. Открытый.</li><li>3. Закрытый.</li><li>4. Таких членов класса не существует.</li></ol>

18. Public – член базового класса при механизме Private-наследования в производном классе имеет доступ

1. Private.
2. Public.
3. Protected.
4. Не наследуется.

19. CASE-средство – это

1. Программное средство, поддерживающее процессы жизненного цикла ПО
2. Язык программирования высокого уровня
3. Система управления базами данных
4. Графический редактор

20. Какой интерфейс пользователя является объектно-ориентированным?

1. Прямого манипулирования.
2. Примитивный.
3. Меню.
4. Со свободной навигацией.

Экзаменатор

\_\_\_\_\_ Тезик К. А.