

УТВЕРЖДАЮ  
Директор ВШТЭ

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

**Б1.Б.18**

(Индекс дисциплины)

**Алгоритмизация и технологии программирования**

(Наименование дисциплины)

Кафедра: **32** Автоматизации технологических процессов и производств

Код

Наименование кафедры

Направление подготовки: 15.03.04 Автоматизация технологических процессов и производств

Профиль подготовки: Автоматизация технологических процессов и производств

Уровень образования: Бакалавриат

### План учебного процесса

Составляющие учебного процесса		Очное обучение	Очно-заочное обучение	Заочное обучение
Контактная работа обучающихся с преподавателем по видам учебных занятий и самостоятельная работа обучающихся (часы)	Всего	<b>144</b>		<b>144</b>
	Аудиторные занятия	<b>51</b>		<b>14</b>
	Лекции	17		6
	Лабораторные занятия			
	Практические занятия	34		8
	Самостоятельная работа	57		121
	Промежуточная аттестация	<b>36</b>		<b>9</b>
Формы контроля по семестрам (номер семестра)	Экзамен	4		5
	Зачет			
	Курсовая работа	4		5
<b>Общая трудоемкость дисциплины (зачетные единицы)</b>		<b>4</b>		<b>4</b>

Форма обучения:	Распределение зачетных единиц трудоемкости по семестрам									
	1	2	3	4	5	6	7	8	9	10
Очная				<b>4</b>						
Очно-заочная										
Заочная					<b>4</b>					

Рабочая программа дисциплины составлена в соответствии с федеральным  
государственным образовательным стандартом высшего образования  
по направлению подготовки 15.03.04 Автоматизация технологических процессов и  
производств

На основании учебных планов № b150304-12\_20  
z150304-12\_20

Кафедра-разработчик: Автоматизации технологических процессов и производств

Заведующий кафедрой: Ковалёв Д.А.

### **СОГЛАСОВАНИЕ:**

Выпускающая кафедра: Автоматизации технологических процессов и производств

Заведующий кафедрой: Ковалёв Д.А.

Методический отдел: Смирнова В.Г.

# 1. ВВЕДЕНИЕ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ

## 1.1. Место преподаваемой дисциплины в структуре образовательной программы

Блок 1: Базовая  Обязательная  Дополнительно является факультативом   
Вариативная  По выбору

## 1.2. Цель дисциплины

Сформировать компетенции обучающегося в области знаний, необходимых для эффективного проектирования средств автоматизации деловых и технологических процессов на основе современных информационных технологий.

## 1.3. Задачи дисциплины

- Рассмотреть понятие алгоритмизации как фундаментальной основы моделирования, проектирования и реализации деловых и производственных процессов и их автоматизации программно-аппаратными средствами вычислительной техники.
- Раскрыть принципы создания программного обеспечения средств автоматизации деловых и технологических процессов.
- Продемонстрировать особенности технологических, инструментальных и лингвистических средств разработки программного обеспечения для автоматизации деловых и технологических процессов и производств.

## 1.4. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Код компетенции	Формулировка компетенции	Этап формирования
ПК-2	Способность выбирать основные и вспомогательные материалы для изготовления изделий, способы реализации основных технологических процессов, аналитические и численные методы при разработке их математических моделей, методы стандартных испытаний по определению физико-механических свойств и технологических показателей материалов и готовых изделий, стандартные методы их проектирования, прогрессивные методы эксплуатации изделий.	2
<b>Планируемые результаты обучения</b> Знать: <ol style="list-style-type: none"><li>1) историю развития парадигм программирования и связанных с ними языков и технологий программирования;</li><li>2) принцип структурной алгоритмизации, базовые алгоритмические структуры и их реализацию в языках программирования высокого уровня;</li><li>3) типовую классификацию данных и их описания в языках программирования высокого уровня;</li><li>4) состав и функции инструментальных средств программирования.</li><li>5) этапы разработки и внедрения интегрированной информационной системы управления предприятием как средства автоматизации его деловых, производственных и технологических процессов.</li></ol> Уметь: <ol style="list-style-type: none"><li>1) самостоятельно разрабатывать и записывать в виде псевдокодов и блок-схем алгоритмы обработки базовых типов данных;</li><li>2) записывать простейшие алгоритмы на алгоритмическом языке программирования высокого уровня, редактировать и отлаживать тексты программ в инструментальной среде программирования;</li><li>3) создавать простейшие приложения для операционной системы Windows, иллюстрирующие технологию визуального программирования;</li><li>4) создавать простейшие программы в технологии объектно-ориентированного программирования.</li></ol> Владеть: <ol style="list-style-type: none"><li>1) навыками структурной алгоритмизации и технологии структурного программирования;</li><li>2) навыками работы в современной визуальной RAD-среде программирования (среда быстрой</li></ol>		

Код компетенции	Формулировка компетенции	Этап формирования
	разработки приложений);	
	3) навыками компонентного проектирования программного обеспечения средств автоматизации.	

### 1.5. Дисциплины (практики) образовательной программы, в которых было начато формирование компетенций, указанных в п.1.4:

- Теория автоматического управления (ПК-2);
- Конструкционные материалы в системах автоматизации (ПК-2).

## 2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Наименование и содержание учебных модулей, тем и форм контроля	Объем (часы)		
	очное обучение	очно-заочное обучение	заочное обучение
<b>Учебный модуль 1. Основы алгоритмизации и программирования</b>			
Тема 1. Понятие алгоритма и его способы записи. Базовые алгоритмические структуры и принцип структурной алгоритмизации. Примеры классических алгоритмов.	17		22
Тема 2. Эволюция программирования как деятельности. Инструменты программирования. Проектирование и внедрение программ.	6		10
<b>Текущий контроль 1.</b> (Компьютерное тестирование)	1		
<b>Учебный модуль 2. Технология алгоритмического программирования</b>			
Тема 3. Понятия алгоритмического программирования. Сущность данных, их классификации для целей программирования. Базовые (простые) и сложные (структурированные) типы данных. Поясняются понятия переменной и константы, их назначение.	14		18
Тема 4. Принципы структурного программирования. Понятия «процедуры» и «функции» как разновидности подпрограмм. Способ взаимодействия подпрограмм посредством передачи параметров при решении общей задачи. Способы описания процедур и функций в теле основной программы.	13		18
<b>Текущий контроль 2.</b> (Компьютерное тестирование)	1		
<b>Учебный модуль 3. Технология событийного программирования</b>			
Тема 5. Понятия визуального программирования. Событийное программирование. Конструирование экранных форм, определение значений свойств элементов управления экранной формы, создание программного кода как совокупность методов обработки событий.	20		24
Тема 6. Объектно-ориентированное программирование. Базовые понятия объекта и класса, их интерпретация как структурированного типа данных. Сравнение с другими структурированными типами данных.	17		22
<b>Текущий контроль 3.</b> (Компьютерное тестирование)	1		3
<b>Курсовая работа</b>	<b>18</b>		<b>18</b>
<b>Промежуточная аттестация по дисциплине (экзамен)</b>	<b>36</b>		<b>9</b>
<b>ВСЕГО:</b>	<b>144</b>		<b>144</b>

## 3. ТЕМАТИЧЕСКИЙ ПЛАН

### 3.1. Лекции

Номера изучаемых тем	Очное обучение		Очно-заочное обучение		Заочное обучение	
	Номер семестра	Объем (часы)	Номер семестра	Объем (часы)	Номер семестра	Объем (часы)
1	4	4			5	1
2	4	2			5	1
3	4	2			5	1
4	4	1			5	1
5	4	4			5	1

Номера изучаемых тем	Очное обучение		Очно-заочное обучение		Заочное обучение	
	Номер семестра	Объем (часы)	Номер семестра	Объем (часы)	Номер семестра	Объем (часы)
6	4	4			5	1
<b>ВСЕГО:</b>		<b>17</b>				<b>6</b>

### 3.2. Практические и семинарские занятия

Номера изучаемых тем	Наименование и форма занятий	Очное обучение		Очно-заочное обучение		Заочное обучение	
		Номер семестра	Объем (часы)	Номер семестра	Объем (часы)	Номер семестра	Объем (часы)
1	Разработка и запись стандартными средствами алгоритмов обработки базовых типов данных (практическая работа).	4	8			5	2
3	Запись разработанных алгоритмов на алгоритмическом языке высокого уровня (практическая работа в компьютерном классе).	4	4			5	1
4	Исследование механизма вызова подпрограмм из основной программы; описание и использование процедур и функций как разновидностей подпрограмм (практическая работа в компьютерном классе).	4	4			5	1
5	Овладение навыками работы в визуальной среде программирования при создании Windows - приложения, освоение приемами обработки некорректных действий пользователя при работе в приложении (практическая работа в компьютерном классе).	4	10			5	2
6	Ознакомление с основными концепциями объектно-ориентированного программирования и исследование типовых операторов и специфических типов данных (объект, класс) языка C++. Проверка области действия внутренних и внешних свойств и методов объектов. (практическая работа в компьютерном классе).	4	8			5	2
<b>ВСЕГО:</b>			<b>34</b>				<b>8</b>

### 3.3. Лабораторные занятия

Не предусмотрено.

## 4. КУРСОВОЕ ПРОЕКТИРОВАНИЕ

### 4.1. Цели и задачи курсовой работы

Цель курсовой работы: приобрести навыки компонентного проектирования и реализации средств автоматизации деловых и технологических процессов.

Задачи курсовой работы:

- реализовать конкретный пример автоматизации делового процесса;
- овладеть навыками работы в современной визуальной RAD-среде программирования;
- овладеть технологией событийного программирования;
- ознакомиться с языком программирования высокого уровня.

### 4.2. Тематика курсовой работы

Курсовая работа включает два уровня заданий:

1. Задание первого уровня: является единым для студентов группы; выполняется на основе методических указаний; не требует глубоких знаний языка программирования; демонстрирует компонентный подход к разработке приложений; представляет собой программный комплекс для тестирования профессиональных компетенций; состоит из двух приложений (APM преподавателя и APM тестируемого), работающих совместно благодаря общим данным, сохраняемым в текстовых файлах; состоит в разработке дизайна экранных форм приложений, задания свойств элементов управления форм, встраивания готовых кодов обработки событий, вызываемых действиями пользователя в экранных формах.

2. Задание второго уровня: является индивидуальным для каждого студента группы; выполняется по аналогии с заданием первого уровня, но без методических указаний, готовых кодов и других подсказок; закрепляет практические навыки в технологии визуального программирования; требует понимания языка программирования в объеме кодов из задания первого уровня.

### 4.3. Требования к выполнению и представлению результатов курсовой работы

Курсовая работа выполняется индивидуально с использованием компьютера с операционной системой Windows и инструментальной визуальной среды программирования Microsoft Visual Basic 6.0. Язык программирования высокого уровня - Visual Basic.

Курсовая работа выполняется в соответствии с методическими указаниями, которые оформлены как отдельный документ обеспечения дисциплины и выдаются студентам в электронной форме. Методические указания включают цели и задачи курсовой работы, этапы ее выполнения, исходные данные, форму представления результатов, варианты индивидуальных заданий.

Результаты представляются в электронном виде как макет (прототип) автоматизированного делового процесса. Результат представляет собой совокупность файлов разработанного проекта, содержащего следующие обязательные элементы:

- папку с файлами выполненного задания первого уровня (файл проекта; файлы экранных форм приложения; файлы модулей приложения);
- папку с файлами выполненного задания второго уровня.

## 5. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ ОБУЧАЮЩЕГОСЯ

Номера учебных модулей, по которым проводится контроль	Форма контроля знаний	Очное обучение		Очно-заочное обучение		Заочное обучение	
		Номер семестра	Кол-во	Номер семестра	Кол-во	Номер семестра	Кол-во
1,2,3	Компьютерное тестирование	4	3			5	1

## 6. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ

Виды самостоятельной работы обучающегося	Очное обучение		Очно-заочное обучение		Заочное обучение	
	Номер семестра	Объем (часы)	Номер семестра	Объем (часы)	Номер семестра	Объем (часы)
Усвоение теоретического материала	4	19			5	55
Подготовка к практическим занятиям	4	20			5	48
Выполнение курсовой работы	4	18			5	18

Виды самостоятельной работы обучающегося	Очное обучение		Очно-заочное обучение		Заочное обучение	
	Номер семестра	Объем (часы)	Номер семестра	Объем (часы)	Номер семестра	Объем (часы)
Подготовка к экзамену	4	36			5	9
	<b>ВСЕГО:</b>					<b>130</b>

## 7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

**7.1. Характеристика видов и используемых инновационных форм учебных занятий**  
Не предусмотрено.

**7.2. Система оценивания успеваемости и достижений обучающихся для промежуточной аттестации**

традиционная  балльно-рейтинговая

## 8. ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 8.1. Учебная литература

а) основная учебная литература

1. Борисенко В.В. Основы программирования [Электронный ресурс]/ Борисенко В.В.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 323 с.— Режим доступа: <http://www.iprbookshop.ru/52206> .— ЭБС «IPRbooks».

б) дополнительная учебная литература

2. Давыдова Н.А. Программирование [Электронный ресурс]: учебное пособие/ Давыдова Н.А., Боровская Е.В.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2015.— 239 с.— Режим доступа: <http://www.iprbookshop.ru/6485>.— ЭБС «IPRbooks».
3. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа: <http://www.iprbookshop.ru/39552>. .— ЭБС «IPRbooks».

**8.2. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине**

1. Андреева Т.А. Программирование на языке Pascal [Электронный ресурс]/ Андреева Т.А.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 277 с.— Режим доступа: <http://www.iprbookshop.ru/52215>.— ЭБС «IPRbooks»
2. Васильев В. Н. Основы программирования на языке С+ [Электронный учебник] : учебное пособие / Васильев В. Н., 2013, Волгоградский институт бизнеса, Вузовское образование. - 72 с. Режим доступа: <http://iprbookshop.ru/11341>.— ЭБС «IPRbooks».

**8.3. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины**

1. Visual Basic для начинающих [Электронный ресурс] URL: <http://www.codenet.ru/progr/vbasic/bit/>
2. Turbo Pascal 7.0. Электронный учебник для студентов [Электронный ресурс] URL: <http://mif.vspu.ru/books/pascal/>
3. Самоучитель С++ с нуля [Электронный ресурс] URL: <https://code-live.ru/tag/cpp-manual/>
4. Материалы Информационно-образовательной среды заочной формы обучения СПбГУПТД [Электронный ресурс]. URL: [http://sutd.ru/studentam/extramural\\_student/](http://sutd.ru/studentam/extramural_student/);

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

1. Microsoft Windows 8.1;
2. Microsoft Office 2013

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

1. Аудитория с мультимедийным комплексом,
2. Компьютерный класс с установленным программным обеспечением, мультимедийным комплексом и выходом в Интернет.

## 8.6. Иные сведения и (или) материалы

1. Мастер-классы и практические занятия проводятся с использованием специализированного программного обеспечения.
2. Конспект в электронном виде.
3. Методические указания в электронном виде.

## 9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Виды учебных занятий и самостоятельная работа обучающихся	Организация деятельности обучающегося
Лекции	<p>Лекции обеспечивают теоретическое изучение дисциплины и проводятся в двух форматах:</p> <ul style="list-style-type: none"><li>• традиционный формат с применением компьютерных презентаций;</li><li>• формат мастер-класса с применением специализированного программного обеспечения.</li></ul> <p>Традиционный формат лекций обеспечивает представление основного содержания дисциплины, обсуждение ретроспективы развития научного направления, позволяет получить комплексное представление о современном состоянии науки в области проектирования программного обеспечения средств автоматизации.</p> <p>Формат мастер-класса позволяет разбирать конкретные примеры применения технологии алгоритмизации и программирования при проектировании и реализации средств автоматизации деловых и технологических процессов.</p> <p>Освоение лекционного материала обучающимся предполагает следующие виды работ:</p> <ul style="list-style-type: none"><li>• проработать рабочую программу в соответствии с целями и задачами, структурой и содержанием дисциплины;</li><li>• составить конспект лекций, кратко, схематично и последовательно фиксируя терминологию, основные положения, выводы</li><li>• составить конспект лекций в формате мастер-класса;</li><li>• проработать опорный конспект дисциплины, ответить на вопросы учебных модулей (п.8.6);</li><li>• работать с основной и дополнительной учебной литературой (п. 8.1);</li><li>• подготовиться и пройти компьютерное тестирование по учебным модулям в режиме offline с доступом к системе тестирования через Интернет.</li></ul>
Практические занятия	<p>На практических занятиях обучающиеся работают в компьютерном классе, овладевают навыками алгоритмизации и технологии программирования, самостоятельного создания алгоритмов и программ, рассматривают конкретные примеры и решают конкретные задачи.</p> <p>Подготовка к практическим занятиям предполагает следующие виды работ:</p> <ul style="list-style-type: none"><li>• работать с конспектом лекций и опорным конспектом;</li><li>• установить на домашнем компьютере специализированное программное обеспечение (п. 8.4);</li><li>• самостоятельно повторять решение разбираемых на мастер-классах практических примеров и задач;</li><li>• работать со справочной информацией из опорного конспекта и информационными ресурсами (п. 8.2 и 8.3).</li><li>• работать с основной и дополнительной учебной литературой в части рассмотрения конкретных примеров алгоритмизации и программирования (п.8.1).</li></ul>



Виды учебных занятий и самостоятельная работа обучающихся	Организация деятельности обучающегося
Самостоятельная работа	<p>Данный вид работы предполагает расширение и закрепление знаний, умений и навыков, усвоенных на аудиторных занятиях. Самостоятельная работа выполняется индивидуально при консультативной поддержке преподавателя.</p> <p>Самостоятельная работа проводится в следующих формах:</p> <ul style="list-style-type: none"> <li>• проработка учебно-методических материалов по дисциплине (п.8.6) и других источников информации (п. 8.1, 8.2, 8.3);</li> <li>• ответы на вопросы для самоконтроля по материалам опорного конспекта (п.8.6)</li> <li>• компьютерное тестирование для самоконтроля по темам дисциплины с использованием Интернет-доступа к системе тестирования;</li> <li>• подготовка к текущему контролю по учебным модулям дисциплины;</li> <li>• выполнение домашних заданий с использованием методических указаний по практическим работам (п.8.6);</li> <li>• выполнение курсовой работы с использованием методических указаний (п.8.6);</li> <li>• подготовка к экзамену, который проводится в форме компьютерного тестирования и выполнения конкретного практического задания, аналогичного рассмотренным на практических занятиях и в домашних заданиях.</li> </ul>

## 10. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### 10.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

#### 10.1.1. Показатели оценивания компетенций на этапах их формирования

Код компетенции (этап освоения)	Показатели оценивания компетенций	Наименование оценочного средства	Представление оценочного средства в фонде
ПК-2 (2)	<p>1. Демонстрирует понимание этапов разработки и внедрения интегрированной информационной системы управления предприятием как средства автоматизации его деловых, производственных и технологических процессов.</p> <p>2. Формулирует принципы структурной алгоритмизации, называет и записывает графически базовые алгоритмические структуры, сопоставляет им описание в языках программирования высокого уровня.</p> <p>3. Записывает простейшие алгоритмы на алгоритмическом языке программирования высокого уровня, редактирует и отлаживает тексты программ в инструментальной среде программирования.</p> <p>4. Владеет навыками работы в современной визуальной среде программирования и навыками компонентного проектирования программного обеспечения средств автоматизации.</p>	<p>1. Тестирование.</p> <p>2. Типовое практическое задание.</p> <p>3. Курсовая работа</p>	<p>1. Тестовые вопросы (150 вопросов).</p> <p>2. Перечень практических заданий (20 заданий).</p> <p>3. Темы и задания к курсовой работе (10 вариантов).</p>

## 10.1.2. Описание шкал и критериев оценивания сформированности компетенций

### Критерии оценивания сформированности компетенций

Оценка по традиционной шкале	Критерии оценивания сформированности компетенций	
	Компьютерное тестирование	Курсовая работа
отлично	<p>Обучающийся свободно ориентируется в основных понятиях, терминах и определениях дисциплины; знает основные принципы структурной алгоритмизации и базовые алгоритмические структуры; умеет самостоятельно разрабатывать и записывать в виде псевдокодов и блок-схем алгоритмы обработки базовых типов данных; умеет записывать алгоритмы на алгоритмическом языке программирования высокого уровня, редактировать и отлаживать тексты программ в инструментальной среде программирования; понимает особенности различных технологий программирования и соответствующих им языков; владеет навыками компонентного проектирования программного обеспечения средств автоматизации; усвоил основную и знаком с дополнительной литературой; проявляет творческие способности и широкую эрудицию в использовании учебного материала.</p> <p>Компьютерное тестирование: результат не менее 86 баллов из 100.</p>	<p>Полное и разностороннее рассмотрение вопросов, свидетельствующее о значительной самостоятельной работе с источниками. Качество исполнения всех элементов работы соответствует требованиям, содержание полностью соответствует заданию. Полученные результаты представлены на основании изучения и анализа исследуемого процесса. Даны исчерпывающие выводы на поставленные вопросы. Работа представлен к защите в требуемые сроки.</p>
хорошо	<p>Обучающийся показывает достаточный уровень знаний основных понятий дисциплины; знает базовые алгоритмические структуры; умеет самостоятельно разрабатывать и записывать в виде блок-схем алгоритмы обработки базовых типов данных; умеет записывать алгоритмы на алгоритмическом языке программирования высокого уровня, редактировать и отлаживать тексты программ в инструментальной среде программирования; усвоил основную литературу; допускает незначительные погрешности при ответах на вопросы.</p> <p>Компьютерное тестирование: результат более 66 баллов, но менее 86 из 100.</p>	<p>Работа выполнен в соответствии с заданием. Имеются отдельные несущественные ошибки в работе или в ответах на поставленные при защите вопросы, могут иметь место отступления от правил оформления работы или нарушены сроки представления работы к защите.</p>
удовлетворительно	<p>Обучающийся показывает знания учебного материала в минимальном объеме; умеет создавать простейшие программы на алгоритмическом языке программирования высокого уровня, редактировать и отлаживать тексты программ в инструментальной среде программирования; может сформулировать основные понятия и определения, но при этом допускает большое количество неприципиальных ошибок; знаком с основной литературой.</p>	<p>Задание выполнено полностью, но в работе есть отдельные существенные ошибки, присутствуют неточности в ответах, либо качество представления работы низкое, либо работа представлена с опозданием.</p>

	Компьютерное тестирование: результат более 56 баллов, но менее 67 из 100.	
неудовлетворительно	Обучающийся не имеет достаточного уровня знания дисциплины; не может сформулировать основные принципы структурной алгоритмизации и программирования; плохо ориентируется в основных понятиях и определениях; не умеет создавать простейшие программы на алгоритмическом языке программирования высокого уровня, редактировать и отлаживать тексты программ в инструментальной среде программирования; плохо знаком с основной литературой; допускает при ответе существенные ошибки и не может устранить их даже под руководством преподавателя. Компьютерное тестирование: результат менее 57 баллов из 100.	Отсутствие одного или нескольких обязательных элементов задания, либо многочисленные грубые ошибки в работе, либо грубое нарушение правил оформления или сроков представления работы. Неспособность ответить на вопрос без помощи экзаменатора.

## 10.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций

### 10.2.1. Перечень вопросов, разработанный в соответствии с установленными этапами формирования компетенций

№ п/п	Формулировка вопросов	№ темы
1	Два определения алгоритма: как фундаментального универсального понятия и как вычислительного процесса. Основные свойства алгоритмов.	1
2	Способы записи алгоритмов. Понятие «исполнение алгоритма» как практическая реализация действий по получению результата для конкретных значений данных и ее запись в табличном виде.	1
3	Принцип структурной алгоритмизации как основа для технологии структурного программирования.	1
4	Базовый набор алгоритмических структур (линейная, ветвящаяся, циклическая) и примеры записи базовых структур формализмом блок-схем.	1
5	Назначение и реализация переменных-счетчиков и переменных-аккумуляторов.	1
6	Алгоритм перестановки значений двух переменных, лежащий в основе алгоритмов сортировки.	1
7	Алгоритмы последовательного и двоичного поиска числа в массиве данных.	1
8	Этапы развития средств вычислительной техники и логических основ ее функционирования.	2
9	Развитие парадигм программирования как совокупности идей и понятий, определяющей стиль написания программ.	2
10	классификация языков программирования по их уровням и принадлежности к парадигмам.	2
11	Программирование как вид деятельности: как наука, как искусство и как ремесло.	2
12	Основные понятия программирования: синтаксис и семантика языка программирования, текст программы, трансляция, тестирование и отладка программы.	2
13	Функции транслятора в процессе создания программы и разновидности трансляторов.	2
14	Структура инструментальной среды для создания программ (системы программирования) и роль ее составляющих. Наиболее популярные системы программирования.	2
15	Этапы создания небольших и средних по объему кода программ.	2
16	Этапы разработки крупного проекта информационной системы.	2
17	Методы маркетинга программного обеспечения.	2
18	Сущность данных, их классификации для целей программирования, базовые (простые) и сложные (структурированные) типы данных.	3
19	Понятия переменной и константы и их назначение.	3

20	Программа как совокупность операторов, предназначенных для целенаправленного преобразования данных. Виды операторов.	3
21	Структура типовой программы, способы описания и инициализации переменных, группировка операторов, способы задания комментариев.	3
22	Сходство операторов, реализующих базовые алгоритмические структуры, в различных языках программирования.	3
23	Структурное программирование как программная реализация структурной алгоритмизации.	4
24	Реализация декомпозиции алгоритма совокупностью подпрограмм, вызываемых из главной программы.	4
25	Понятия «процедура» и «функция» как разновидности подпрограмм.	4
26	Способ взаимодействия подпрограмм посредством передачи параметров при решении общей задачи.	4
27	Понятие «визуальное программирование» для разработки windows-приложений.	5
28	Windows-приложения как окна (экранные формы) с набором элементов управления.	5
29	Действия пользователя как события, на которые реагирует windows-приложение.	5
30	Объектно-ориентированное программирование как технология конструирования приложений на базе объектных репозиториях.	6
31	Базовые понятия объекта и класса и их интерпретация как структурированного типа данных.	6
32	Сравнение понятий объекта и класса с другими структурированными типами данных.	6
33	Способы описания класса, инициализации объекта как экземпляра класса.	6
34	Три основные концепции объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.	6

#### 10.2.2. Вариант тестовых заданий, разработанных в соответствии с установленными этапами формирования компетенций

№ п/п	Формулировка задания	Ответ
1	Программа для процессора представляет собой: а). текст на языке программирования высокого уровня б). алгоритм решения задачи в). машинный код г). состояние электрических элементов д). логику предикатов	в
2	Основным дискретом оперативной памяти является в том числе: а). диод б). транзистор в). байт г). фраза д). поле	в
3	Алгоритм - это: а). инструкция решения задачи с бесконечным множеством исходов б). список альтернативных вариантов решения задачи в). программа, написанная на языке высокого уровня г). конечная последовательность действий решения задачи д). программа, написанная в машинных кодах	г
4	Свойство алгоритма «дискретность» подразумевает, что: а). описываемый процесс и его алгоритм могут быть разбиты на отдельные элементарные этапы б). алгоритм не зависит от состояния начальных данных и предпочтений разработчика в). предписания алгоритма должны быть точными и понятными г). алгоритм должен приводить к получению результата за конечное число шагов д). алгоритм пригоден для решения всех задач данного типа	а
5	Псевдокод записи алгоритма - это: а). словесное описание последовательности действий естественным языком б). описание последовательности действий специальными кодами в). словесное описание последовательности действий на подмножестве естественного языка г). описание последовательности действий специальным шрифтом д). описание последовательности действий специальными шрифтом	в

6	В алгоритм может входить одна из следующих алгоритмических структур: а). реляционная б). клиент-серверная в). файловая г). динамическая д). ветвящаяся	д
7	Текст программы - это: а). детальное описание алгоритма на языке программирования б). блок-схема алгоритма в). последовательность двоичных чисел г). запись алгоритма псевдокодами д). запись условия задачи на естественном языке	а
8	Семантика языка - это а). правила написания слов языка б). смысл слов и конструкций языка в). способ перевода слов на другой язык г). правила расстановки знаков препинания в тексте д). правила произношения слов	б
9	Алгоритмическое программирование - это: а). разбиение программы на последовательность модулей, каждый из которых выполняет одно или несколько действий б). выполнение отдельных модулей программы в асинхронном режиме на различных компьютерах сети в). выполнение модулей программы в синхронном режиме на различных компьютерах сети г). правила написания программы на языке программирования высокого уровня д). правила написания программы на языке программирования низкого уровня	а
10	Переменная в программировании указывает на: а). значение, связанное с конкретным типом данных б). алгоритм обработки события, связанного с данными в). функцию вывода результата вычислений г). функцию ввода данных в программу д). вектор прерывания с конкретным адресом стека	а
11	Массив - это: а). сложный тип данных, доступ к элементам которого происходит по их именам б). базовый тип данных, доступ к элементам которого происходит по имени в). сложный тип данных, образованный сочетанием различных базовых типов г). сложный тип данных, доступ к элементам которого происходит по их номерам (индексам) д). простой тип данных, состоящий из индексов	г
12	Инициализация переменной в программировании - это а). присвоение переменной начального значения б). вычисление значения переменной в результате выполнения арифметической операции в). резервирование памяти под хранение величины заданного типа г). удаление переменной из программы д). приравнивание значения переменной нулю	а
13	Условный оператор служит для: а). реализации многократно повторяющихся действий б). разветвления последовательности выполнения программы по результату проверки условия в). резервирования памяти под новую переменную г). выполнения арифметических действий над переменными д). выполнения циклических операций	б
14	Арифметические операции применимы к: а). строковым данным б). логическим данным в). целым и вещественным числовым данным г). символьным данным д). всем типам данных	в

15	<p>Функция как разновидность подпрограмм в языке программирования играет следующую роль:</p> <p>а). управляет ходом выполнения главной программы</p> <p>б). выполняет логически законченную последовательность действий</p> <p>в). обеспечивает ввод данных в оперативную память</p> <p>г). вычисляет значение параметра и возвращает его в вызывающую программу</p> <p>д). обеспечивает вывод данных на стандартное устройство вывода</p>	г
----	--	---

### 10.2.3. Вариант типовых заданий, разработанных в соответствии с установленными этапами формирования компетенций

№ п/п	Условия типовых заданий	Ответ
1	Разработать и записать в виде блок-схем алгоритмы решения следующей задачи: ввести одномерный массив целых чисел, определить его максимальный элемент.	<pre> graph TD     Start([Начало]) --&gt; Data[Список данных N=5, A[N] - целые i, max - целые]     Data --&gt; Input[/Ввод: A[N]/]     Input --&gt; Init1[max := A[1]]     Init1 --&gt; Init2[i:=2]     Init2 --&gt; Loop{A[i] &gt; max}     Loop -- + --&gt; Update1[max := A[i]]     Update1 --&gt; Update2[i := i+1]     Update2 --&gt; Loop     Loop -- - --&gt; Exit{ }     Exit --&gt; LoopEnd{i &gt; N}     LoopEnd -- + --&gt; Output[/Вывод: max/]     Output --&gt; End([Конец]) </pre>
2	Используя оператор цикла, написать и отладить программу: ввести одномерный массив целых чисел, определить его максимальный элемент, результат вывести на экран.	<pre> Program massiv_1; Uses CRT; Const n=10; Var A: array[1..n] of integer; i: byte; max: integer; BEGIN   ClrScr;   write ('Введите элементы массива: ');   for i:=1 to n do     readln (A[i]);     max:=A[1];   for i:=2 to n do     if A[i] &gt; max then       max:=A[i];   write ('наибольший элемент массива = ', max); END. </pre>

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

#### 10.3.1. Условия допуска обучающегося к сдаче экзамена и защите курсовой работы и порядок ликвидации академической задолженности

Положение о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся.

### 10.3.2. Форма проведения промежуточной аттестации по дисциплине

устная  письменная  компьютерное тестирование  иная

Проверка теоретических знаний обучающегося проводится в форме компьютерного тестирования, проверка практических знаний проводится в письменной форме. Результаты проверки доводятся до обучающегося в течение 30 минут после завершения экзамена.

### 10.3.3. Особенности проведения экзамена и защиты курсовой работы

1. Экзамен по дисциплине проводится в компьютерном классе с выходом в Интернет в форме компьютерного тестирования и выполнения практического задания.
2. Тестовые вопросы выбираются автоматически случайным образом из единой базы данных вопросов по дисциплине. В базе данных содержится 150 тестовых вопросов.
3. Вопросы имеют три степени сложности и оцениваются соответственно на 3 (легкие), 4 (средней сложности) и 5 (сложные) баллов.
4. В каждом тесте содержится 45 вопросов трех категорий сложности. На выполнение теста отводится 20 минут.
5. При условии успешного прохождения тестирования обучающийся получает индивидуальное практическое задание, для выполнения которого может потребоваться компьютер (практические задания, связанные с разработкой и отладкой программ).
6. Время выполнения практического задания – 20 минут.
7. Во время экзамена студенту запрещено пользоваться какими-либо электронными устройствами и литературой в любом формате. Допускается наличие чистых листов бумаги и пишущих принадлежностей.

### Особенности защиты курсовой работы

1. Защита курсовой работы проводится в компьютерном классе, оснащенном соответствующим программным обеспечением.
2. Форма представления результатов курсовой работы к защите – электронная.
3. Результат представляет собой совокупность файлов разработанного прототипа автоматизированного делового процесса «АРМ преподавателя» и «АРМ тестируемого», организованных в две папки (файлы проектов; файлы экранных форм приложения; файлы модулей приложения).
4. В процессе защиты обучающийся демонстрирует работоспособность разработанных приложений, имитируя действия пользователей этих АРМов.
5. Для подтверждения приобретенных компетенций в области технологии визуального программирования обучающемуся предлагается выполнить одно из типовых индивидуальных заданий второго уровня.
6. Критерием оценки курсовой работы служит полнота выполнения индивидуального задания в установленный регламентом срок – 1 час.