

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Выборнова Ирина Александровна
Должность: Профессор
Дата подписания: 03.02.2022 15:17:47
Уникальный программный ключ:
c3b3b9c625f6c115afa2a2c42ba1f9e05a38b76e

МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СЕРВИСА» (ФГБОУ ВО «ПВГУС»)

Кафедра «Информационный и электронный сервис»

РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА

по дисциплине «Технология программирования»

для студентов специальности 09.02.01 «Компьютерные системы и комплексы»

Тольятти 2018 г.

Рабочая учебная программа по дисциплине «Технология программирования» включена в основную профессиональную образовательную программу специальности 09.02.01 «Компьютерные системы и комплексы» решением Президиума Ученого совета

Протокол № 4 от 28.06.2018 г.

Начальник учебно-методического отдела _____  Н.М.Шемендюк
28.06.2018 г.


Рабочая учебная программа по дисциплине «Технология программирования» разработана в соответствии с Федеральными государственными образовательными стандартами: специальности 09.02.01 «Компьютерные системы и комплексы» утвержденный приказом Министерства образования и науки Российской Федерации от 28.07.2014 №849.

Составил: к.физ-мат.н, ст.пр. Е.С. Устинова

СОГЛАСОВАНО:

Директор научной библиотеки _____  В.Н.Еремина

СОГЛАСОВАНО:


Начальник управления информатизации _____  В.В.Обухов

Рабочая программа утверждена на заседании кафедры «Информационный и электронный сервис»

Протокол № 11 от «27» июня 2018 г.

Заведующий кафедрой _____  д.т.н., профессор В.И. Воловач

СОГЛАСОВАНО:

Начальник учебно-методического отдела _____  Н.М.Шемендук

1. Перечень планируемых результатов обучения по дисциплине (модулю, междисциплинарному курсу), соотнесенных с планируемыми результатами освоения образовательной программы

1.1. Цели освоения дисциплины (модуля, междисциплинарного курса)

Целями освоения дисциплины (модуля, междисциплинарного курса) являются: основных этапов решения задач на ЭВМ; критериев качества программы; диалоговых программ; дружелюбности, жизненного цикла программы; постановки задачи и спецификации программы; способов записи алгоритма; программ на языке высокого уровня; стандартных типов данных; представления основных структур программирования: итерация, ветвление, повторение, процедуры; типов данных, определяемых пользователем, записей, файлов, динамических структур данных; списков: основные виды и способы реализации; программирования рекурсивных алгоритмов; способов конструирования программ; модульных программ; основ доказательства правильности.

1.2. Результаты освоения (личностные, метапредметные, предметные) в соответствии с ФГОС среднего общего образования обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины у обучающихся формируются следующие компетенции:

Код компетенции	Наименование компетенции
ПК 1.3	Использовать средства и методы автоматизированного проектирования при разработке цифровых устройств.
ПК 2.1	Создавать программы на языке ассемблера для микропроцессорных систем.

1.3. Перечень планируемых результатов обучения по дисциплине

Результаты освоения дисциплины	Технологии формирования компетенции по указанным результатам	Средства и технологии оценки по указанным результатам
Знает: методы автоматизированного проектирования при разработке цифровых устройств ПК 1.3; навыки алгоритмического мышления и понимает необходимость формального описания алгоритмов ПК 2.1.	Лекции, лабораторные, практические занятия	Собеседование
Умеет: Использовать методы автоматизированного проектирования при разработке цифровых устройств ПК 1.3; понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня ПК 2.1.	Лекции, лабораторные, практические занятия	Тестирование
Имеет практический опыт: Использовать средства и методы автоматизированного проектирования при разработке цифровых устройств ПК 1.3; создавать программы на языке высокого уровня для микропроцессорных систем ПК	Написание контрольной работы	Собеседование

2.1.		
------	--	--

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к вариативной части цикла ОП. Ее освоение осуществляется в 6, 7 семестрах* у студентов очной формы обучения и в 5, 6 семестрах* у студентов заочной формы обучения.

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Код и наименование компетенции(й)
	Предшествующие дисциплины (практики)	
	Основы алгоритмизации и программирования	<p>ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес. ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность. ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития. ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности. ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями. ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий. ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации. ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.</p> <p>ПК 2.1. Создавать программы на языке ассемблера для микропроцессорных систем ПК 2.2. Производить тестирование, определение параметров и отладку микропроцессорных систем. ПК 3.3. Принимать участие в отладке и технических испытаниях компьютерных систем и комплексов, инсталляции, конфигурировании программного обеспечения.</p>
	Операционные системы и среды	<p>способность: ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес. ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность. ОК 4. Осуществлять</p>

		<p>поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития. ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности. ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями. ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий. ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации. ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.</p> <p>ПК 2.3. Осуществлять установку и конфигурирование персональных компьютеров и подключение периферийных устройств.</p> <p>ПК 3.3. Принимать участие в отладке и технических испытаниях компьютерных систем и комплексов, инсталляции, конфигурировании программного обеспечения.</p> <p>ПК 4.3. Проводить мероприятия по защите информации в компьютерных системах и комплексах.</p>
Последующие дисциплины (практики)		
	Программное обеспечение компьютерных сетей и web-серверов	<p>ПК 1.3 Использовать средства и методы автоматизированного проектирования при разработке цифровых устройств.</p> <p>ПК 2.3 Осуществлять установку и конфигурирование персональных компьютеров и подключение периферийных устройств.</p> <p>ПК 3.1 Проводить контроль параметров, диагностику и восстановление работоспособности компьютерных систем и комплексов.</p> <p>ПК 3.2 Проводить системотехническое обслуживание компьютерных систем и комплексов.</p> <p>ПК 3.3 Принимать участие в отладке и технических испытаниях компьютерных систем и комплексов, инсталляции, конфигурировании программного обеспечения..</p>

*Здесь и далее семестры указаны для обучающихся на базе основного общего образования. Для лиц, обучающихся на базе среднего общего образования, семестры соответствуют учебному плану и нормативному сроку обучения, установленному ФГОС.

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

Распределение фонда времени по семестрам и видам занятий

Виды занятий	очная форма обучения	очно-заочная форма обучения	заочная форма обучения
Итого часов Зачетных единиц	244 ч.	_____ ч.	244 ч.
Лекции (час)	72		6
Практические (семинарские) занятия (час)	78		8
Лабораторные работы (час)	42		4
Самостоятельная работа (час)	52		220
Курсовой проект (работа) (+,-)	-		-
Контрольная работа (+,-)	+		+
Экзамен, семестр /час.	-		-
Зачет (дифференцированный зачет), семестр	7		6
Контрольная работа, семестр	6		5

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

4.1. Содержание дисциплины

№ п/п	Раздел дисциплины	Виды учебных занятий, включая самостоятельную работу студентов и трудоемкость (в академических часах)				Средства и технологии оценки
		Лекции, час	Практические (семинарские) занятия, час	Лабораторные работы, час	Самостоятельная работа, час	
1	Тема 1 Приемы обеспечения технологичности программных продуктов.	12/1	6/2		8/36	Конспект, сообщение/ защита практической работы
2	Тема 2 Анализ требований и определение спецификаций программного обеспечения при структурном подходе.	12/1		8/2	8/36	Конспект, сообщение/ защита лабораторных работ
3	Тема 3 Проектирование программного обеспечения при структурном подходе. Анализ требований и определение спецификаций программного	12/1	12/2		10/36	Конспект, сообщение/ защита практических работ

	обеспечения при объектном подходе.					
4	Тема 4 Рабочий поток определения требований. Моделирование прецедентов. Рабочий поток анализа. Объекты и классы. Пакеты анализа. Реализация прецедентов. Диаграммы деятельности.	12/1		34/2	10/36	Конспект, сообщение/ защита лабораторных работ
5	Тема 5 Проектирование программного обеспечения при объектном подходе.	12/1	6		6/36	Конспект, защита практических работ
6	Тема 6 Простейшие программы с экранной формой и элементами управления. Инициирование и обработка событий мыши и клавиатуры. Чтение, запись текстовых и бинарных файлов, текстовый редактор. Исключения, шаблоны и новые возможности Visual C++. Редактирование графических данных. Управление буфером обмена с данными в текстовом и графическом форматах	12/1	54/4		10/40	Конспект, защита практических работ, контрольная работа
	Промежуточная аттестация по дисциплине	72/6	78/8	42/4	52/220	дифференцированный зачет, контрольная работа

4.2.Содержание практических (семинарских) занятий

№	Наименование темы практических (семинарских) занятий	Объем часов	Форма проведения
6/5 семестр			
1	Занятие 1. « Сортировка по ключу одномерных массивов»	6/2	решение разноуровневых и проблемных задач, защита практических работ
2	Занятие 2. « Поиск по ключу в одномерном массиве структур»	6	решение разноуровневых и проблемных задач, защита практических работ
3	Занятие 3. «Программирование с использованием однонаправленных списков типа «стек»»	6	решение разноуровневых и проблемных задач, защита практических работ
4	Занятие 4. Разработка простейшего Windows-приложения.	6/2	решение разноуровневых и проблемных задач, защита практических работ
5	Занятие 5. Программирование с использованием файлов.		решение разноуровневых и проблемных задач, защита практических работ
6	Занятие 6. Разработка многооконного приложения.	6	решение разноуровневых и проблемных задач, защита

			практических работ
7	Занятие 7. Разработка программ с элементами графики.	6	решение разноуровневых и проблемных задач, защита практических работ
8	Занятие 8. Создание меню в Visual C++.	4	решение разноуровневых и проблемных задач, защита практических работ
Итого за 6/5 семестр		46/4	
7/6 семестр			
1	Занятие 9. Разработка простейшего интернет-приложения.	8/2	решение разноуровневых и проблемных задач, защита практических работ
2	Занятие 10. Взаимодействие C++ с Microsoft Office.	8	решение разноуровневых и проблемных задач, защита практических работ
3	Занятие 11. Разработка программы для обработки базы данных.	8/2	решение разноуровневых и проблемных задач, защита практических работ
4	Занятие 12. Создание XML-документа методами C++.	8	решение разноуровневых и проблемных задач, защита практических работ
Итого за 7/6 семестр		32/4	
Итого			

4.3. Содержание лабораторных работ

№	Наименование лабораторных работ	Объем часов	Наименование темы дисциплины
6/5 семестр			
1	Лабораторная работа 1. «Разработка требований к системе»	4/2	Анализ требований и определение спецификаций программного обеспечения при структурном подходе.
2	Лабораторная работа 2. «Основы работы в Rational Rose 2003. Создание диаграммы прецедентов (вариантов использования)»	6	Рабочий поток определения требований. Моделирование прецедентов.
3	Лабораторная работа 3. «Создание диаграмм классов и состояния»	4	Рабочий поток анализа. Объекты и классы. Пакеты анализа. Реализация прецедентов.
4	Лабораторная работа 4. «Создание диаграмм действий»	6	Диаграммы деятельности.
Итого за 6/5 семестр		22ч./ 2ч.	
7/6 семестр			
1	Лабораторная работа 5. «Создание проекта в MS Project 2007»	4/2	Ввод и вывод табличных данных. Решение системы уравнений.
2	Лабораторная работа 6. «Создание связей между задачами в MS Project 2007»	4	Использование функций MS Word, MS Excel, AutoCAD и MATLAB, а также создание PDF-файла.
3	Лабораторная работа 7. «Создание списка	4	Использование технологии LINQ

	ресурсов в MS Project 2007»		
4	Лабораторная работа 8. «Другие подходы к ресурсному планированию в MS Project 2007»	4	Отладка программного обеспечения.
5	Лабораторная работа 9. «Управление рисками в MS Project 2007»	4	Составление программной документации.
	Итого за 7/6 семестр	20ч./2ч.	
	Итого	42ч./4ч.	

5. Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине

Технологическая карта самостоятельной работы студента

Вид деятельности студентов (задания на самостоятельную работу)	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объем часов
1	2	3	4
Выполнение индивидуальных заданий в виде реферата на заданную тему.	реферат	собеседование	2/30
Выполнение индивидуальных заданий в виде реферата на заданную тему.	реферат	собеседование	2/30
Итого за 6/5 семестр			4/60
Выполнение индивидуальных заданий в виде реферата на заданную тему.	реферат	собеседование	24/80
Выполнение индивидуальных заданий в виде реферата на заданную тему.	реферат	собеседование	24/80
Итого за 7/6 семестр			48/160
Итого			52/220

Литература:

1. Голицына, О. Л. Языки программирования [Электронный ресурс] : учеб. пособие для сред. проф. образования / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - 3-е изд., перераб. и доп. - Документ Bookread2. - М. : Форум [и др.], 2015. - 398 с. - Библиогр.: с. 363-364. - Глоссарии терминов. - Режим доступа: <http://znanium.com/bookread2.php?book=493421>
2. Иванова, Г. С. Технология программирования [Текст] : учеб. для вузов по направлению "Информатика и вычисл. техника" / Г. С. Иванова. - 3-е изд., стереотип. - М. : КноРус, 2016. - 334 с. : ил. - Библиогр.: с. 329-331.

Содержание заданий для самостоятельной работы

В разделе «Самостоятельная работа» раскрывается содержание каждого вида самостоятельной работы и указывается время, необходимое для его выполнения.

Самостоятельная работа студента по дисциплине включает в себя:

1. Изучение лекционного материала по конспекту лекций.
2. Подготовку к лабораторным занятиям.

Преподаватель в соответствии с рабочей учебной программой дисциплины доводит до студентов: перечень лабораторных работ, последовательность их выполнения, рекомендуемые учебно-методические пособия, руководства и другую информацию, необходимую для подготовки к практическому выполнению предусмотренных программой дисциплины лабораторных работ.

При подготовке к лабораторным работам студент использует рекомендованные учебники и учебные пособия, руководства по выполнению лабораторных работ, инструкции по пользованию измерительной аппаратурой, а также специальные указания по особенностям выполнения отдельных пунктов лабораторных работ.

Подготовка к лабораторным и практическим работам осуществляется студентами самостоятельно заблаговременно.

В процессе такой подготовки студент должен усвоить теоретический материал, относящийся к данной лабораторной и практической работе, изучить и ясно представить себе содержание и порядок выполнения лабораторной и практической работ.

Темы рефератов

- 1) Специфические особенности программных систем.
- 2) Особенности современных методологий и технологий разработки программных продуктов.
- 3) Технология структурного программирования.
- 4) Технология сборочного программирования.
- 5) Направления развития и модели концепции открытых систем.
- 6) Технология объектно-ориентированного программирования.
- 7) Технология применения CASE-систем.
- 8) Состав, структура и функциональные особенности CASE-средств.
- 9) Особенности и возможности Internet-технологии.
- 10) Особенности и возможности Intranet-технологии.

Вопросы для самоконтроля

1. Технология программирования и основные этапы ее развития.
2. Проблемы разработки сложных программных систем.
3. Блочный-иерархический подход к созданию сложных систем.
4. Жизненный цикл и этапы разработки программного обеспечения.
5. Эволюция моделей жизненного цикла программного обеспечения.
6. Ускорение разработки программного обеспечения.
7. Технология RAD.
8. Оценка качества процессов создания программного обеспечения.
9. Понятие технологичности программного обеспечения.
10. Модули и их свойства.
11. Нисходящая и восходящая разработка программного обеспечения.
12. Структурное и «неструктурное» программирование.
13. Средства описания структурных алгоритмов.
14. Стиль оформления программы.
15. Эффективность и технологичность.
16. Программирование «с защитой от ошибок».
17. Сквозной структурный контроль.
18. Классификация программных продуктов по функциональному признаку.
19. Основные эксплуатационные требования к программным продуктам.
20. Предпроектные исследования предметной области.

6. Методические указания для обучающихся по освоению дисциплины Инновационные образовательные технологии

Вид образовательных технологий, средств передачи знаний, формирования умений и практического опыта	№ темы / тема лекции	№ практического (семинарского) занятия/наименование темы	№ лабораторной работы / цель
Слайд-лекция, собеседование	1. Приемы обеспечения технологичности	Занятие 1. Сортировка по ключу одномерных	

	программных продуктов.	массивов.	
Слайд-лекция, собеседование	2. Анализ требований и определение спецификаций программного обеспечения при структурном подходе.		Лабораторное занятие № 1. Цель работы: Освоить методы составления технического задания на проектирование программного продукта.
Слайд-лекция, собеседование	3. Проектирование программного обеспечения при структурном подходе.	Занятие 2. Поиск по ключу в одномерном массиве структур	
Слайд-лекция, собеседование	4. Анализ требований и определение спецификаций программного обеспечения при объектном подходе.	Занятие 3. Программирование с использованием однонаправленных списков типа «стек».	
Слайд-лекция, собеседование	5. Рабочий поток определения требований. Моделирование прецедентов.		Лабораторное занятие № 2. Цель работы: Освоить основы моделирования диаграмм «вариантов использования» в CASE-системе StarUML.
Слайд-лекция, собеседование	6. Рабочий поток анализа. Объекты и классы. Пакеты анализа. Реализация прецедентов.		Лабораторное занятие № 3. Цель работы: Освоить основы моделирования диаграмм классов и состояния в CASE-системе StarUML.
Слайд-лекция, собеседование	7. Диаграммы деятельности.		Лабораторное занятие № 4. Цель работы: Создание диаграмм действий.
Слайд-лекция, собеседование	8. Простейшие программы с экранной формой и элементами управления.	Занятие 4. Разработка простейшего Windows-приложения	
Слайд-лекция, собеседование	9. Чтение, запись текстовых и бинарных файлов, текстовый редактор.	Занятие 5. Программирование с использованием файлов.	
Слайд-лекция, собеседование	10. Исключения, шаблоны и новые возможности Visual C++ .	Занятие 6. Разработка многооконного приложения.	
Слайд-лекция, собеседование	11. Редактирование графических данных.	Занятие 7. Разработка программ с элементами графики.	
Слайд-лекция, собеседование	12. Управление буфером обмена с данными в текстовом и	Занятие 8. Создание меню в Visual C++.	

	графическом формате.		
Слайд-лекция, собеседование	13. Ввод и вывод табличных данных. Решение системы уравнений.		Лабораторное занятие № 5. Цель работы: Освоить методы разработки проектов в MS Project 2007.
Слайд-лекция, собеседование	14. Элемент управления WebBrowser.	Занятие 9. Разработка простейшего интернет-приложения.	
Слайд-лекция, собеседование	15. Использование функций MS Word, MS Excel, AutoCAD и MATLAB, а также создание PDF-файла.	Занятие 10. Взаимодействие C++ с Microsoft Office.	Лабораторное занятие № 6. Цель работы: Научиться создавать проект со связями между задачами.
Слайд-лекция, собеседование	16. Обработка баз данных с использованием технологии ADO.NET.	Занятие 11. Разработка программы для обработки базы данных.	
Слайд-лекция, собеседование	17. Использование технологии LINQ	Занятие 12. Создание XML-документа методами C++.	Лабораторное занятие № 7. Цель работы: Научиться создавать списки ресурсов с использованием MS Project 2007.
Слайд-лекция, собеседование	18. Отладка программного обеспечения.		Лабораторное занятие № 8. Цель работы: Освоить методы расчета ресурсов и трудозатрат по требуемому проекту.
Слайд-лекция, собеседование	19. Составление программной документации.		Лабораторное занятие № 9. Цель работы: Научиться создавать проекты по управлению рисками в MS Project 2007.

В начале семестра студентам необходимо ознакомиться с технологической картой дисциплины, выяснить, какие результаты освоения дисциплины заявлены (знания, умения, практический опыт). Для успешного освоения дисциплины студентам необходимо выполнить задания, предусмотренные рабочей учебной программой дисциплины и пройти контрольные точки в сроки, указанные в технологической карте (раздел 11). От качества и полноты их выполнения будет зависеть уровень сформированности компетенции и оценка текущей успеваемости по дисциплине. По итогам текущей успеваемости студенту может быть выставлена оценка по промежуточной аттестации, если это предусмотрено технологической картой дисциплины. Списки учебных пособий, научных трудов, которые студентам следует прочесть и законспектировать, темы практических занятий и вопросы к ним, вопросы к экзамену (зачету) и другие необходимые материалы указаны в разработанном для данной дисциплины учебно-методическом комплексе.

Основной формой освоения дисциплины является контактная работа с преподавателем - лекции, практические занятия, лабораторные работы (при наличии в учебном плане),

консультации (в том числе индивидуальные), в том числе проводимые с применением дистанционных технологий.

По дисциплине часть тем (разделов) изучается студентами самостоятельно. Самостоятельная работа предусматривает подготовку к аудиторным занятиям, выполнение заданий (письменных работ, творческих проектов и др.) подготовку к промежуточной аттестации (экзамену (зачету)).

На лекционных и практических (семинарских) занятиях вырабатываются навыки и умения обучающихся по применению полученных знаний в конкретных ситуациях, связанных с будущей профессиональной деятельностью. По окончании изучения дисциплины проводится промежуточная аттестация (экзамен, (зачет)).

Регулярное посещение аудиторных занятий не только способствует успешному овладению знаниями, но и помогает организовать время, т.к. все виды учебных занятий распределены в семестре планомерно, с учетом необходимых временных затрат.

6.1. Методические указания для обучающихся по освоению дисциплины на лабораторных работах

Лабораторные работы

№	Наименование лабораторных работ	Задание по лабораторным работам
1	Лабораторная работа 1. «Разработка требований к системе»	По своему варианту разработать концепцию, глоссарий и дополнительные спецификации проекта. Вариант 1. Разработать программу для администрирования гостиницы. В программе должна храниться информация о проживающих клиентах, номерах и служащих гостиницы. Администратор гостиницы может добавлять, удалять и изменять эту информацию. Ему могут потребоваться следующие сведения: список работников гостиницы; список свободных номеров с указанием его вместимости; прейскурант цен; справка о жильцах гостиницы (ФИО, срок проживания, номер); отчет о работе гостиницы за месяц. Вариант 2. Разработать информационную систему поддержки поставок скоропортящихся продуктов. Вариант 3. Разработать информационную систему деятельности склада.
2	Лабораторная работа 2. «Основы работы в Rational Rose 2003. Создание диаграммы прецедентов (вариантов использования)»	Разработать Диаграмму вариантов использования для системы. Вариант задания взять из лабораторной работы №1. Отчет должен содержать главную Диаграмму вариантов использования, описание актеров и их деятельности, описание потоков событий для главной диаграммы.
3	Лабораторная работа 3. «Создание диаграмм классов и состояния»	Разработать Диаграмму классов и диаграмму состояния для одного из классов. Вариант задания взять из лабораторной работы №1. Отчет должен содержать описание классов, диаграмму классов, описание создания диаграммы классов, диаграмму состояний и описание процесса ее создания.
4	Лабораторная работа 4. «Создание диаграмм действий»	Разработать диаграмму действий и диаграмму взаимодействия для одного из вариантов использования. Вариант задания взять из лабораторной работы №1. Для диаграммы действий описать потоки событий, создать диаграмму для главного потока событий и одного под-потока событий.

		Для диаграммы взаимодействия описать основные объекты, таблицу сообщений между ними и составить диаграмму.
5	Лабораторная работа 5. «Создание проекта в MS Project 2007»	Создайте подробную диаграмму Ганта для перечисленных ниже систем. Планирование производить по методике планирования от даты начала проекта. Сохраните созданный проектный файл для дальнейшего использования. 1. Управление структурой факультета вуза. 2. Управление дисциплинами специальности в вузе. 3. Управление студенческими группами кафедры. 4. Терминал для покупки билетов в театральной кассе. 5. Управление заказами и корзиной Интернет – магазина. 6. Управление размещением клиентов в гостинице.
6	Лабораторная работа 6. «Создание связей между задачами в MS Project 2007»	Оформить связи между задачами в проекте по варианту из лабораторной работы №5.
7	Лабораторная работа 7. «Создание списка ресурсов в MS Project 2007»	Определить для своего варианта работы из лабораторной работы №5 используемые ресурсы. Назначить ресурсы задачам.
8	Лабораторная работа 8. «Другие подходы к ресурсному планированию в MS Project 2007»	Создать для своего варианта работы из лабораторной работы №5 пул ресурсов. Назначить дополнительные ресурсы тем задачам, где основные ресурсы перегружены.
9	Лабораторная работа 9. «Управление рисками в MS Project 2007»	Для ранее созданных проектов выполнить следующие виды анализа: а) PERT-анализ длительностей задач. При этом самостоятельно задать в таблице ввода PERT значения оптимистической, ожидаемой и пессимистической длительностей. б) Параметрический анализ длительностей задач. При этом нужно самостоятельно назначить некоторые задачи как параметрические; создать настраиваемые поля для параметра, нормы и оценки длительности; ввести значения параметра и нормы длительности и вычислить оценку длительности.

Лабораторные работы обеспечивают:

формирование умений и навыков обращения с приборами и другим оборудованием, демонстрацию применения теоретических знаний на практике, закрепление и углубление теоретических знаний, контроль знаний и умений в формулировании выводов, развитие интереса к изучаемой дисциплине.

Применение лабораторных работ позволяет вовлечь в активную работу всех обучающихся группы и сформировать интерес к изучению дисциплины.

Самостоятельный поиск ответов на поставленные вопросы и задачи в ходе лабораторной работы приобретают особую значимость в восприятии, понимании содержания дисциплины.

Изученный на лекциях материал лучше усваивается, лабораторные работы демонстрируют практическое их применение.

6.2. Методические указания для выполнения контрольных работ

Контрольная работа - одна из форм проверки и оценки усвоенных знаний, получения информации о характере познавательной деятельности, уровне самостоятельности и активности учащихся в учебном процессе, об эффективности методов, форм и способов учебной деятельности.

Для выполнения контрольной работы студенты должны изучить теоретические основы C++. В ходе выполнения контрольной работы приобретается опыт в разработке программ с четкой структуризацией, умение осуществлять постановку задачи, применять процедуры и функции из стандартных модулей при создании проекта, конструировать собственные подпрограммы, использовать данные в виде файлов.

Для успешного выполнения контрольной работы необходимо изучить рекомендуемую и дополнительную литературу параллельно с прослушиванием лекций для закрепления материала.

Рекомендуется следующий порядок работы:

- ознакомиться по учебной программе с содержанием темы;
- прочитать раздел учебника и другую техническую литературу, относящуюся к данной теме;
- изучить конспект лекции по темам;
- решить задачу с использованием конкретного задания.

Пояснительная записка должна состоять из введения, основной части и заключения.

В **введении** необходимо определить цель выполнения заданий контрольной работы, их основную идею и наметить пути достижения поставленной цели. Объем введения 1-2 страницы.

Основная часть пояснительной записки должна содержать:

- формулировку задачи;
- UML диаграммы;
- описание входных, выходных и промежуточных данных;
- листинг программы;
- исходные данные для тестирования (контрольный пример) с полученным

результатом.

Объем основной части составляет 10-20 страниц.

В **заключение** пояснительной записки к контрольной работе делаются краткие выводы о полученных результатах, оценивается оптимальность решения задачи. Объем заключения - 1-2 страницы.

Пояснительная записка **оформляется** на листах бумаги формата А4. Текстовая часть оформляется на принтере на одной стороне листа с соблюдением полей: сверху и снизу 20 мм, слева 35 мм, справа 10 мм, через 1,5 межстрочных интервала. Текст пояснительной записки рамкой не обводится. Страницы пояснительной записки **нумеруются** подряд в **верхнем правом углу** страницы. Нумерация страниц начинается с титульного листа, номер на котором не проставляется. Структурная схема программы выполняется на компьютере. Заголовки разделов пишутся заглавными буквами, заголовки подразделов - строчными буквами с “красной” строки. Перенос слов в заголовках не допускается. Точки в конце заголовков разделов и подразделов не ставятся. Расстояние между заголовками и текстовой частью должно составлять 15 мм. Каждый новый раздел следует начинать с нового листа. Текст пояснительной записки должен быть кратким, содержательным и грамотным. Сокращения слов в тексте недопустимы.

В список литературы включаются все используемые в работе источники. Сведения о книгах включают фамилии и инициалы авторов, заглавие книги, место издания, издательство, год издания, количество страниц. Никаких кавычек нигде не ставится. Города Москва и Ленинград, как место издания, указываются сокращенно М., Л. Названия других городов пишутся полностью. Если на титульном листе книги не указан автор, то сведения начинают с указания названия книги, затем после наклонной черты (/) и слов “Под ред.” указываются инициалы и фамилия редактора и далее данные в той же последовательности, что и выше.

Контрольная работа предполагает решение задач на языке программирования C++.

К **стилю программирования** предъявляются следующие требования:

- Программа должна быть удобочитаема, простая и ясная;
- Использовать в качестве идентификаторов переменных осмысленные имена;
- Записывать только один оператор в строке;
- Использовать сдвиги в строке в соответствии с уровнем вложенности;
- Использовать комментарии в тексте программы.

Задание:

- 1) Разработать 4 UML диаграммы по заданному варианту.
- 2) Разработать программу по заданному варианту с визуальным представлением.

Для успешного выполнения контрольной работы необходимо изучить рекомендуемую и дополнительную литературу параллельно с прослушиванием лекций для закрепления материала.

Рекомендуется следующий порядок работы:

- ознакомиться по учебной программе с содержанием темы;
- прочитать раздел учебника и другую техническую литературу, относящуюся к данной теме;
- изучить конспект лекции по темам;
- решить задачу с использованием конкретного задания.

№ темы/тема	примерная тематика для выполнения контрольных работ	задания
Тема 6. Простейшие программы с экранной формой и элементами управления. Инициирование и обработка событий мыши и клавиатуры. Чтение, запись текстовых и бинарных файлов, текстовый редактор. Исключения, шаблоны и новые возможности Visual C++. Редактирование графических данных. Управление буфером обмена с данными в текстовом и графическом форматах	<p>1. Список товаров, имеющих на складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести список товаров, хранящихся больше месяца и стоимость которых превышает 1 000 000 р.</p> <p>2. Для получения места в общежитии формируется список студентов, который включает ФИО студента, группу, средний балл, доход на члена семьи. Вывести информацию о студентах, у которых доход на члена семьи менее двух минимальных зарплат.</p> <p>3. В справочной автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, пункт назначения, время отправления и прибытия. Вывести информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени.</p>	<p>Разработать 4 UML диаграммы по заданному варианту.</p> <p>Разработать программу по заданному варианту с визуальным представлением.</p>

7. Фонда оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине «Технология программирования» зачет

Фонды оценочных средств, позволяющие оценить уровень сформированности и результаты освоения дисциплины, представлены следующими компонентами:

Код оцениваемой компетенции (или ее части)	Тип контроля	Вид контроля	Количество элементов
ПК 1.3, ПК 2.1	текущий	устный опрос	1-30
	текущий	тест	20-60
ПК 1.3, ПК 2.1	текущий	письменный ответ	1-30
	промежуточный	тест	1-60

7.1. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

Результаты освоения дисциплины	Оценочные средства (перечень вопросов)
--------------------------------	--

Знает: методы автоматизированного проектирования при разработке цифровых устройств ПК 1.3; навыки алгоритмического мышления и понимает необходимость формального описания алгоритмов ПК 2.1.

I: UML диаграммы

S: В языках UML набор согласованных ролей, которые могут играть пользователи при взаимодействии с системой:

-: класс

-: интерфейс

+: актер

-: компонент

I: UML-диаграммы

S: В языках UML набор операций, которые определяют услуги класса или компонента:

-: класс

+: интерфейс

-: актер

-: компонент

I: UML-диаграммы

S: В языках UML физическая и заменяемая часть системы, которая соответствует набору интерфейсов и обеспечивает реализацию этого набора интерфейсов:

-: класс

-: интерфейс

-: актер

+: компонент

I: UML-диаграммы

S: Диаграмма взаимодействия, которая выделяет упорядочение сообщений по времени

-: диаграмма классов

-: диаграмма объектов

-: диаграмма кооперации

+: диаграмма последовательности

I: UML-диаграммы

S: В языках UML описание последовательности действий (или нескольких последовательностей), выполняемых системой в интересах отдельного актера и производящих видимый для актера результат

-: класс

-: интерфейс

-: актер

+: Элемент Use Case

I: UML-диаграммы

S: Набор классов, интерфейсов, сотрудничеств и их отношений

+: диаграмма классов

-: диаграмма объектов

-: диаграмма кооперации

-: диаграмма последовательности

I: Алгоритм и его свойства

S: Строгая и четкая система правил, которая определяет последовательность действий над некоторыми объектами и после конечного числа шагов приводит к достижению поставленной цели – это...?

-: алгоритмическая система

+: алгоритм

-: алгоритмизация

-: аналитика

I: Ассоциация и наследование

S: Какой тип мощности ассоциации бывает?
-: многие к одному
+: и тот, и другой
-: один к – одному
I: Ассоциация и наследование
S: Выберите ассоциацию с типом мощность один ко многим:
+: клиент – заказ
-: человек – здание
-: студент – преподаватель
-: европейская жена – европейский муж
I: Ассоциация и наследование
S: Выберите ассоциацию с типом мощность многие ко многим:
-: клиент – заказ
-: человек – здание
-: студент – преподаватель
+: Б и В
I: Ассоциация и наследование
S: Тип наследования, при котором производный класс имеет одного родителя
+: простое
-: сложное
-: множественное
-: одинарное
I: Ассоциация и наследование
S: Тип наследования, при котором производный класс имеет несколько базовых
-: простое
-: сложное
+: множественное
-: одинарное
I: Ассоциация и наследование
S: Замена реализации функции в базовом классе на реализацию в производном – это...
-: обновление
+: переопределение
-: перегрузка
-: рекурсия
I: Взаимоотношения объектов
S: Семантическое отношение между двумя предметами, в котором изменение в одном предмете (независимом предмете) может влиять на семантику другого предмета (зависимого предмета)
+: зависимость
-: ассоциация
-: агрегация
-: реализация
I: Виды алгоритмов
S: Укажите базовую структуру псевдокода?
-: Следствие
-: Вилка
+: Цикл
-: Начало
I: Данные

S: Данные – это:

+: любая информация, пригодная для обработки алгоритмом

-: такие данные, значения которых могут изменяться в процессе выполнения алгоритма

-: данные, значения которых не меняются в процессе выполнения алгоритма

-: нет верного ответа

I: Документация разработчика

S: Документ, в котором сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки и регламентирован процесс приемно-сдаточных испытаний:

-: пояснительная записка

+: техническое задание

-: ГОСТ

-: спецификация

I: Документы программирования

S: Документ, в котором сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки и регламентирован процесс приемно-сдаточных испытаний:

-: пояснительная записка

+: техническое задание

-: ГОСТ

-: спецификация

I: Документы разработчика

S: Документ, в котором сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки и регламентирован процесс приемно-сдаточных испытаний

-: пояснительная записка

+: техническое задание

-: ГОСТ

-: спецификация

I: Жизненный цикл ПО

S: Назовите модель жизненного цикла, предполагающая последовательно расположенные этапы, начинающиеся только после того, как закончатся предыдущие, результат в конце разработки

-: спиральная модель

-: итерационная модель

+: каскадная модель

-: нет верного ответа

I: Жизненный цикл ПО

S: Назовите модель жизненного цикла, предполагающая разработку новых версий ПО:

+: спиральная модель

-: итерационная модель

-: каскадная модель

-: нет верного ответа

I: Какими свойствами должен обладать алгоритм?

S: Какими свойствами должен обладать алгоритм?

-: однозначность, правильность, неопределенность

-: правильность, неопределенность, массовость

+: массовость, правильность, однозначность

I: Классы и объекты

S: Из каких частей состоит класс:

+: название, атрибуты, операции

-: идентификатор, атрибуты, операции

-: атрибуты, операции, адрес

-: название, атрибуты, значения

I: Классы и объекты

S: В определении класса члены класса с ключевым словом `private` доступны

-: любой функции программы

-: в случае, если вам известен пароль

+: методам
этого класса

-: только открытым членам класса

I: Классы и объекты

S: Операция точки (операция доступа к члену класса) объединяет следующие два элемента (слева направо):

-: член класса и объект класса

-: объект класса и класс

-: класс и член этого класса

+: объект класса и член этого класса

I: Классы и объекты

S: Методу класса всегда доступны данные:

+: объекта, членом которого он является

+: класса, членом которого он является

-: любого объекта класса, членом которого он является

-: класса, объявленного открытым

I: Классы и объекты

S: Выберите верное обращение к переменной `Year` объекта `Old` класса `Car`:

+: `Old.Year = 20`

-: `Car.Year = 20`

-: `Year.Old = 20`

-: `Year.Old = 20`

I: Классы и объекты

S: В определении класса члены класса с ключевым словом `public` доступны:

+: любой функции программы

-: составным и дружественным функциям классов, которые являются производными от этого класса или совпадают с ним

-: методам этого класса

-: только открытым членам класса

I: Классы и объекты

S: В определении класса члены класса с ключевым словом `protected` доступны:

-: любой функции программы

+: составным и дружественным функциям классов, которые являются производными от этого класса или совпадают с ним

-: методам этого класса

-: только открытым членам класса

I: конструирование ПО

S: Компания, в которой процесс конструирования ПО и

принимаемые решений зависят только от таланта конкретных разработчиков, называется ...

-: зрелой

+: незрелой

-: простой

-: неразвитой

I: Методы класса

S: Метод класса, осуществляющий последовательный доступ к элементам данных, определяющих состояние класса:

-: конструктор

-: модификатор

-: деструктор

+: итератор

I: Методы класса

S: Если все методы какого-либо класса должны иметь доступ к скрытым полям другого, весь класс объявляется...

-: константным

-: статическим

+: дружественным

-: нет верного ответа

I: Методы классов

S: Метод класса, служащий для уничтожения экземпляров класса:

-: конструктор

-: модификатор

+: деструктор

-: итератор

I: Методы классов

S: Метод класса, служащий для создания экземпляров класса:

+: конструктор

-: модификатор

-: деструктор

-: итератор

I: Методы классов

S: Метод класса, служащий для обработки состояния класса без его изменения:

-: конструктор

-: модификатор

-: деструктор

+: селектор

I: Методы классов

S: Укажите конструктор для класса Car:

+: Car

-: ~Car

-: destructor Car

-: нет верного ответа.

I: Методы классов

S: Выберите верное объявление константной функции:

-: void const Jump()

-: const void Jump()

+: void Jump() const

-: нет верного ответа

I: Модели зрелости компании

S: Укажите уровень модели зрелости компании, при которой главной задачей компании становится постоянное улучшение и повышение эффективности существующих процессов, ввод новых технологий.

-: начальный

-: повторяемый

-: определенный

+: оптимизирующий

I: Модели зрелости компании

S: Укажите уровень модели зрелости компании, при которой в компании принимаются количественные показатели качества как программных продуктов, так и процесса.

-: начальный

+: управляемый

-: определенный

-: оптимизирующий

I: Модели зрелости компании

S: Укажите уровень модели зрелости компании, при которой процесс не может строго планироваться и отслеживаться, его успех носит случайный характер.

+: начальный

-: управляемый

-: определенный

-: оптимизирующий

I: Назовите этапы жизненного цикла программного продукта:

S: Назовите этапы жизненного цикла программного продукта:

-: планирование, реализация, сопровождение

-: тестирование,

эксплуатация, передача в свободное пользование

-: программирование, анализ требований, отладка

+: планирование, реализация, сопровождение

и тестирование, эксплуатация, передача в свободное пользование

I: Основы программирования

S: Процесс, при котором содержимое текстового файла преобразуется в исполняемый машинный код, называется?

-: компоновка

+: компиляция

-: обработка

-: структурирование

I: Отношения между классами

S: Отношения между классами

-: ассоциация, обобщение-специализация

-: зависимость, рациональность

-: целое-часть

+: ассоциация, обобщение-специализация и целое-часть

I: Отношения между классами

S: Отношение, при котором один класс разделяет структуру и поведение, определенные в одном другом или во многих других классах – это?

-: метакласс
 -: ассоциация
 +: наследование
 -: реализация
 I: Отношения между объектами
 S: Структурное отношение, которое описывает набор связей, являющихся соединением между объектами:
 -: зависимость
 +: ассоциация
 -: агрегация
 -: реализация
 I: Переменные
 S: Переменные – это:
 -: любая информация, пригодная для обработки алгоритмом
 +: такие данные, значения которых могут изменяться в процессе выполнения алгоритма
 -: данные, значения которых не меняются в процессе выполнения алгоритма
 -: нет верного ответа
 I: Принципы объектно-ориентированного программирования
 S: Формирование представления о качествах или свойствах предмета путем мысленного удаления некоторых частей или материальных объектов – это...
 +: абстракция
 -: инкапсуляция
 -: наследование
 -: полиморфизм
 I: Принципы объектно-ориентированного программирования
 S: Объединение данных с функциями их обработки в сочетании со скрытием ненужной для использования этих данных информации – это...
 -: абстракция
 +: инкапсуляция
 -: наследование
 -: наследование
 I: Принципы объектно-ориентированного программирования
 S: Возможность создания иерархии классов, когда потомки наследуют все свойства своих предков, могут их изменять и добавлять новые – это...
 -: абстракция
 +: инкапсуляция
 -: наследование
 -: полиморфизм
 I: Принципы объектно-ориентированного программирования
 S: Основными свойствами объектно-ориентированного программирования являются:
 +: абстракция, инкапсуляция, полиморфизм
 -: инкапсуляция, полиморфизм, однозначность
 -: наследование, массовость, инкапсуляция
 -: полиморфизм, абстракция, определенность

	<p>I: Процедуры работы компаний S: Компания, в которой работают ясные процедуры управления проектами и построения программных продуктов, называется... +: зрелой -: незрелой -: сложной -: развитой</p>
<p>Умеет: Использовать методы автоматизированного проектирования при разработке цифровых устройств ПК 1.3; понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня ПК 2.1.</p>	<p>I: Рекурсивная подпрограмма S: Рекурсивная подпрограмма – это: +: подпрограмма, которая обращается сама к себе -: подпрограмма, которая повторяется определенное количество раз -: главная подпрограмма -: нет верного ответа I: Свойства алгоритмов S: Какими свойствами должен обладает алгоритм? +: дискретность, результативность, определенность -: правильность, неопределенность, результативность -: массовость, конечность, неоднозначность -: нет верного ответа I: Свойства алгоритмов S: Что означает свойство дискретности алгоритма? -: Единственность толкования правил выполнения действий и порядка их выполнения +: Решение задачи, записанное в виде алгоритма, должно быть разбито на отдельные простейшие шаги, которые расположены в порядке их выполнения -: Получение конечного результата за конечное время -: Неопределенность результатов выполнения каждого шага алгоритма за конечное время I: Свойства алгоритмов S: Что означает свойство определенности алгоритма? -: Возможность использования алгоритма для некоторого класса исходных данных. -: Однозначная определенность результатов выполнения каждого шага алгоритма за конечное время +: Однозначная определенность результатов выполнения каждого шага алгоритма. -: Решение задачи, записанное в виде алгоритма, должно быть разбито на отдельные простейшие шаги, которые расположены в порядке их выполнения. I: Свойства алгоритмов S: Что означает свойство формальности? -: Решение задачи, записанное в виде алгоритма, должно быть разбито на отдельные простейшие шаги, которые расположены в порядке их выполнения -: Неопределенность результатов выполнения каждого шага алгоритма за конечное время +: Любой исполнитель, способный воспринимать и выполнять инструкции алгоритма отвлекается от содержания поставленной задачи и лишь строго выполняет инструкции</p>

-: Возможность использования алгоритма для некоторого класса исходных данных

I: Сортировка и поиск

S: Процесс перестановки объектов данного множества в определенном порядке:

+ : сортировка

-: поиск

-: перенос

-: обмен

I: Сортировка и поиск

S: Укажите две категории сортировки:

+ : внутренняя и внешняя

-: внутренняя и межфайловая

-: внешняя и межфайловая

-: нет верного ответа

I: Сортировка и поиск

S: Сортировка, при которой на каждом этапе выбирается элемент с наибольшим значением ключа и меняется местами с последним элементом, называется:

-: метод «пузырька»

-: сортировка простым включением

+ : сортировка простым выбором

-: сортировка Хоора

I: Сортировка и поиск

S: Сортировка, при которой все элементы разделяют на упорядоченную и неупорядоченную части, очередной элемент из начала неупорядоченной вставляется на подходящее место в упорядоченную часть, называется?

-: метод «пузырька»

+ : сортировка простым включением

-: сортировка простым выбором

-: сортировка Хоора

I: Сортировка и поиск

S: Сортировка массива, при которой используется «метод просеивания вверх», называется:

-: метод «пузырька»

-: сортировка простым включением

-: сортировка Хоора

+ : сортировка с помощью двоичного дерева

I: Сортировка и поиск

S: Поиск в оперативной памяти, в таблице, называется:

-: внешний

+ : внутренний

-: программный

-: нет верного ответа

I: Сортировка и поиск

S: Метод поиска, при котором из таблицы с данными формируется двоичное дерево:

-: поиск с использованием Хэш-таблицы

-: последовательный поиск

+ : логарифмический поиск

-: простой поиск

I: Сортировка и поиск

S: Метод поиска, при котором таблица с данными

преобразуется в новую таблицу с ключами, созданными

специальной функцией, называется:

+: поиск с использованием Хэш-таблицы

-: последовательный поиск

-: логарифмический поиск

-: простой поиск

I: Стратегии конструирования ПО

S: Назовите методы относящиеся к экстремальному программированию:

+: парное программирование

-: бизнес-моделирование

-: бизнес-моделирование

-: нет верного ответа

I: Стратегии конструирования ПО

S: Процессы, которые ориентированы на человеческие качества и требующие меньший объем документации, называются:

+: облегченными

-: тяжеловесными

-: экстремальными

-: экстремальными

I: Структуры данных

S: Упорядоченная структура однотипных данных – это:

-: запись

-: файл

+: массив

-: нет верного ответа

I: Структуры данных

S: Массив – это...

+: область памяти, где могут размещаться совокупности данных определенного типа

-: область данных, где могут размещаться совокупности памяти определенного типа

-: область памяти, где могут размещаться совокупности памяти определенного типа

-: область данных, где могут размещаться совокупности значений определенного типа

I: Структуры данных

S: Константа – это:

-: любая информация, пригодная для обработки алгоритмом

-: такие данные, значения которых могут изменяться в процессе выполнения алгоритма

+: данные, значения которых не меняются в процессе выполнения алгоритма

-: нет верного ответа

I: Структуры данных

S: Объединение элементов произвольных типов – это:

+: запись

-: файл

-: массив

-: нет верного ответа

I: Структуры данных

S: Логически непрерывный именованный набор данных на внешнем носителе:

-: запись

+: файл
 -: массив
 -: жесткий диск
 I: Структуры данных
 S: Список, работающий по принципу: «последним пришел – первым вышел»?
 +: стек
 -: очередь
 -: дерево
 -: бинарное дерево
 I: Структуры данных
 S: Список, работающий по принципу: «первым пришел – первым ушел»?
 -: стек
 +: очередь
 -: дерево
 -: бинарное дерево
 I: Структуры данных
 S: Логически непрерывный именованный набор данных на внешнем носителе:
 -: запись
 +: файл
 -: массив
 -: жесткий диск
 I: Структуры данных
 S: Структура, в которой помимо данных хранятся также адреса элементов?
 -: запись
 -: файл
 +: список
 -: массив
 I: Структуры программ
 S: Укажите структуру, которая предусматривает повторное выполнение определенных действий, необходимое для большинства программ.
 -: Вилка
 -: Следствие
 +: Цикл
 -: Развилка
 I: Структуры программ
 S: Какая структура означает, что два действия должны быть выполнены друг за другом?
 -: Следствие
 +: Следование
 -: Цикл
 -: Шаговая
 I: Структуры программ
 S: Укажите структуру цикла "от до"
 -: цикл от $I := N1$ до $N2$
 < условие >
 кц
 -: цикл от < действие > до < условия >
 < действие >
 кц

-: цикл от *< действие >* до *< условия >*
< действие >

кц

+: цикл от *I := N1* до *N2*
< действие >

кц

I: Структуры программ

S: Укажите структуру, которая обеспечивает выбор одной из двух альтернатив?

-: Вилка

-: Следствие

-: Цикл

+: Развилка

I: Технология конструирования программного обеспечения

S: Система инженерных принципов для создания экономичного ПО, которое надежно и эффективно работает в реальных компьютерах – это?

-: система конструирования программного обеспечения

+: технология конструирования программного обеспечения

-: процедура конструирования программного обеспечения

-: нет верного ответа

I: Технология конструирования программного обеспечения

S: Решения каких задач обеспечивают методы ТКПО?

+: кодирование

-: формирование отчетов, форм по соответствующим требованиям

-: макетирование

-: нет верного ответа

I: Технология конструирования программного обеспечения

S: Что является «клеем», который соединяет методы и утилиты так, что они обеспечивают непрерывную технологическую цепочку разработки?

-: средства ТКПО

-: методы ТКПО

+: процедуры ТКПО

-: нет верного ответа

I: Технология конструирования программного обеспечения

S: Что процедуры ТКПО определяют?

-: кодирование

+: формирование отчетов, форм по соответствующим требованиям

-: макетирование

-: нет верного ответа

I: Технология конструирования программного обеспечения

S: Укажите популярные парадигмы ТКПО?

-: кодирование

-: формирование отчетов, форм по соответствующим требованиям

+: макетирование

-: нет верного ответа

I: Типы алгоритмов

S: Наглядное графическое изображение алгоритма, когда

отдельные его действия изображаются при помощи различных геометрических фигур, а связи между этапами указываются при помощи стрелок, соединяющих эти фигуры?

- : график
- : диаграмма
- : структурограмма
- +: схема

I: Типы алгоритмов

S: Какие типы алгоритмов различают в зависимости от применяемых базовых структур?

- +: линейные, разветвляющиеся
- : комбинированные, замкнутые
- : циклические

I: Типы данных

S: Тип данных, описывающий значения «истина» и «ложь»?

- : вещественный
- : символьный
- +: логический
- : целый

I: Типы данных

S: Тип данных, описывающий цифры, буквы, знаки препинания и т.д.?

- : вещественный
- +: символьный
- : логический
- : целый

I: Типы программ

S: Рекурсивная подпрограмма – это:

- +: подпрограмма, которая обращается сама к себе
- : подпрограмма, которая повторяется определенное количество раз
- : главная подпрограмма
- : главная подпрограмма

I: Типы файлов

S: Тип файлов для которого характерны операции чтения и записи по произвольному адресу

+: файл прямого доступа

- : файл последовательного доступа
- : файл произвольного доступа
- : файл косвенного доступа

I: Формы записи алгоритмов

S: Укажите существующие формы записи алгоритмов?

- : бинарная
- +: псевдокод
- : запись функциональная;

I: Функции и модули

S: Деление программы на функции и модули является основой:

- : процедурного программирования

	+: структурного программирования -: объектно-ориентированного программирования
<p>Имеет практический опыт: Использовать средства и методы автоматизированного проектирования при разработке цифровых устройств ПК 1.3; создавать программы на языке высокого уровня для микропроцессорных систем ПК 2.1.</p>	<p>I: Характеристики объектов S: Характеристика объекта, которая отличает его от всех других объектов – это?</p> <p>-: поведение -: состояние +: индивидуальность -: нет верного ответа</p> <p>I: Характеристики объектов S: Это у объекта характеризуется перечнем всех свойств объекта и текущими значениями каждого из этих свойств?</p> <p>-: поведение +: состояние -: индивидуальность -: нет верного ответа</p> <p>I: Характеристики объектов S: Укажите отличие активного объекта от пассивного? -: активный объект может изменять своё состояние только под воздействием других объектов -: не имеет собственного канала управления +: имеет собственный канал управления</p> <p>I: Характеристики объектов S: Физическое или понятийное соединение между объектами – это? -: коннект +: связь -: соединение -: соединение</p> <p>I: Характеристики объектов S: Какие роли объект не может играть? -: Актер -: Сервер -: Агент +: Все три</p> <p>I: Характеристики объектов S: Объект, который может воздействовать на другие объекты, но никогда не подвержен воздействию других объектов – это... +: актер -: сервер -: агент -: диктатор</p> <p>I: языки UML S: В языках UML описание множества объектов, которые разделяют одинаковые свойства, операции, отношения и семантику (смысл): +: класс -: интерфейс -: актер -: компонент</p> <p>I: Языки высокого уровня делятся на S: Языки высокого уровня делятся на</p>

	-: машинные, машинно-ориентированные, машинно-независимые +: процедурные, логические, объектно-ориентированные -: машинные, ассемблеры, процедурные -: объектно-ориентированные, логические, машинные
--	--

7.2. Методические рекомендации к определению процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Рабочая учебная программа дисциплины содержит следующие структурные элементы:

- перечень компетенций, формируемых в результате изучения дисциплины с указанием этапов их формирования в процессе освоения образовательной программы;
- 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 (далее – задания). Задания по каждой компетенции, как правило, не должны повторяться.

Требования по формированию задания на оценку ЗНАНИЙ:

- обучающийся должен воспроизводить и объяснять учебный материал с требуемой степенью научной точности и полноты;
- применяются средства оценивания компетенций: тестирование, вопросы по основным понятиям дисциплины и т.п.

Требования по формированию задания на оценку УМЕНИЙ:

- обучающийся должен решать типовые задачи (выполнять задания) на основе воспроизведения стандартных алгоритмов решения;
- применяются следующие средства оценивания компетенций: простые ситуационные задачи (задания) с коротким ответом или простым действием, упражнения, задания на соответствие или на установление правильной последовательности, эссе и другое.

Требования по формированию задания на оценку навыков и (или) ОПЫТА ДЕЯТЕЛЬНОСТИ:

- обучающийся должен решать усложненные задачи (выполнять задания) на основе приобретенных знаний, умений и навыков, с их применением в определенных ситуациях;
- применяются средства оценивания компетенций: задания требующие многошаговых решений как в известной, так и в нестандартной ситуациях, задания, требующие поэтапного решения и развернутого ответа, ситуационные задачи, проектная деятельность, задания расчетно-графического типа. Средства оценивания компетенций выбираются в соответствии с заявленными результатами обучения по дисциплине.

Процедура выставления оценки доводится до сведения обучающихся в течение месяца с начала изучения дисциплины путем ознакомления их с технологической картой дисциплины, которая является неотъемлемой частью рабочей учебной программы по дисциплине.

В результате оценивания компетенций на различных этапах их формирования по дисциплине студенту начисляются баллы по шкале, указанной в рабочей учебной программе по дисциплине.

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

Успешность усвоения дисциплины характеризуется качественной оценкой на основе листа оценки сформированности компетенций, который является приложением к зачетно-экзаменационной ведомости при проведении промежуточной аттестации по дисциплине.

Критерии оценивания компетенций

Компетенция считается сформированной, если теоретическое содержание курса освоено полностью; при устных собеседованиях студент исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты;

проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 86 до 100, что соответствует *повышенному уровню* сформированности компетенции.

Компетенция считается сформированной, если теоретическое содержание курса освоено полностью; при устных собеседованиях студент последовательно, четко и логически стройно излагает учебный материал; справляется с задачами, вопросами и другими видами заданий, требующих применения знаний; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 61 до 85,9, что соответствует *пороговому уровню* сформированности компетенции.

Компетенция считается несформированной, если студент при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет практические работы, не демонстрирует необходимых умений, доля невыполненных заданий, предусмотренных рабочей учебной программой составляет 55 %, качество выполненных заданий не соответствует установленным требованиям, качество их выполнения оценено числом баллов ниже 61, что соответствует *допороговому уровню*.

Шкала оценки уровня освоения дисциплины

Качественная оценка может быть выражена: в процентном отношении качества усвоения дисциплины, которая соответствует баллам, и переводится в уровневую шкалу и оценки «отлично» / 5, «хорошо» / 4, «удовлетворительно» / 3, «неудовлетворительно» / 2, «зачтено», «не зачтено». Преподаватель ведет письменный учет текущей успеваемости студента в соответствии с технологической картой по дисциплине.

Шкала оценки результатов освоения дисциплины, сформированности компетенций

Шкалы оценки уровня сформированности компетенции (й)		Шкала оценки уровня освоения дисциплины		
<i>Уровневая шкала оценки компетенций</i>	<i>100 балльная шкала, %</i>	<i>100 балльная шкала, %</i>	<i>5-балльная шкала, дифференцированная оценка/балл</i>	<i>недифференцированная оценка</i>
допороговый	ниже 61	ниже 61	«неудовлетворительно» / 2	не зачтено
пороговый	61-85,9	70-85,9	«хорошо» / 4	зачтено
		61-69,9	«удовлетворительно» / 3	зачтено
повышенный	86-100	86-100	«отлично» / 5	зачтено

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Списки основной литературы

1. Голицына, О. Л. Языки программирования [Электронный ресурс] : учеб. пособие для сред. проф. образования / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - 3-е изд., перераб. и доп. - Документ Bookread2. - М. : Форум [и др.], 2015. - 398 с. - Режим доступа: <http://znanium.com/bookread2.php?book=493421>.

2. Иванова, Г. С. Технология программирования [Текст] : учеб. для вузов по направлению "Информатика и вычисл. техника" / Г. С. Иванова. - 3-е изд., стереотип. - М. : КноРус, 2016. - 334 с.

3. Учебно-методический комплекс по дисциплине "Технология программирования" [Электронный ресурс] : для студентов специальности 09.02.01 "Компьютерные системы и комплексы" / Поволж. гос. ун-т сервиса (ФГБОУ ВПО "ПВГУС"),

Каф. "Информ. и электрон. сервис" ; сост. М. С. Данилова. - Документ Adobe Acrobat. - Тольятти : ПВГУС, 2015. - 6,10 МБ, 294 с. - Режим доступа: <http://elib.tolgas.ru>.

Списки дополнительной литературы

4. Введение в СУБД MySQL [Электронный ресурс] : учеб. курс. – Режим доступа: <http://www.intuit.ru/department/database/mysql>.

5. Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы [Электронный ресурс] : учеб. для студентов техн. специальностей / В. А. Гвоздева. - Документ Bookread2. - М. : ФОРУМ [и др.], 2015. - 541 с. - Режим доступа: <http://znanium.com/bookread2.php?book=492670#>.

6. Иванова, Г. С. Программирование [Текст] : учеб. для вузов по направлению 230100 "Информатика и вычисл. техника" / Г. С. Иванова. - 3-е изд., стер. - М. : КноРус, 2014. - 432 с.

7. Сергеева, И. И. Информатика [Электронный ресурс] : учеб. для сред. проф. образования / И. И. Сергеева, А. А. Музалевская, Н. В. Тарасова. - 2-е изд., перераб. и доп. - Документ Bookread2. - М. : ФОРУМ [и др.], 2017. - 384 с. - Режим доступа: <http://znanium.com/bookread2.php?book=768749>.

8.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет"), необходимых для освоения дисциплины

Интернет-ресурсы

1. Википедия [Электронный ресурс]. - Режим доступа: <http://ru.wikipedia.org/>. - Загл. с экрана.

2. ИНТУИТ. Национальный Открытый Университет [Электронный ресурс]. - Режим доступа: <http://www.intuit.ru/>. - Загл. с экрана.

3. Электронная библиотечная система Поволжского государственного университета сервиса [Электронный ресурс]. - Режим доступа: <http://elib.tolgas.ru/>. - Загл. с экрана.

4. Электронно-библиотечная система Znanium.com [Электронный ресурс]. - Режим доступа: <http://znanium.com/>. - Загл. с экрана.

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

Краткая характеристика применяемого программного обеспечения

№ п/п	Программный продукт	Характеристика	Назначение при освоении дисциплины
1	CASE-системе StarUML.	Рабочий поток определения требований. Моделирование прецедентов.	Язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.
2	Microsoft Project (или MSP) —	Ввод и вывод табличных данных. Решение системы уравнений. Элемент управления WebBrowser. Отладка программного обеспечения. Составление программной документации.	Программа управления проектами

3	Visual C++	Управление буфером обмена с данными в текстовом и графическом форматах.	Интегрированная среда разработки приложений на языке C++
---	------------	---	--

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

Реализация программы дисциплины в соответствии с требованиями ФГОС СПО по специальности требует наличие учебного кабинета, укомплектованного специализированной мебелью, техническими средствами обучения, и лаборатории программирования, оснащенной лабораторным оборудованием различной степени сложности.

11. Примерная технологическая карта дисциплины _____

Поволжский государственный университет сервиса

Факультет СПО

Технологическая карта дисциплины Технология программирования

Кафедра «Информационный и электронный сервис», преподаватель _____

Группы _____, семестр осенний 20__-20__ учебного года

№	Виды контрольных точек	Количество контрольных точек	Количество баллов за 1 контрольную точку	Срок прохождения контрольных точек																Итого	Зачетно-экзаменационная сессия		
				сентябрь					октябрь				ноябрь				декабрь						
				1	7	14	21	28	5	12	19	26	2	9	16	23	30	7	14			21	28
				6	13	20	27	4	11	18	25	1	8	15	22	29	6	13	20	27	3		
1	Обязательные:																						
1.1	Посещение лекционных занятий	11	1		+	+	+	+	+	+	+	+	+	+								11	
1.2	Защита лабораторных работ	6	5					+		+				+		+						30	
1.3	Выполнение контрольной работы	1	10																+			10	
2	Творческий рейтинг:																						
2.1	Участие в студенческой конференции	1	15											+								15	
3	Контрольное тестирование:																						
3.1	Промежуточное тестирование	1	14										+									14	
3.2	Итоговое тестирование	1	20																	+		20	
4	Форма контроля																						
	Итого																					100	

1) При условии выполнения всех обязательных контрольных точек студент может получить 61 балл.

2) Для всех контрольных точек указано максимальное количество баллов.

Поволжский государственный университет сервиса
Факультет СПО
Технологическая карта дисциплины Технология программирования
Кафедра «Информационный и электронный сервис», преподаватель _____
Группы _____, семестр 2 20__-20__ учебного года

№	Виды контрольных точек	Количество контрольных	Количество баллов за 1	Срок прохождения контрольных точек																		Итого	Зачетно-экзаменационная сессия			
				январь		февраль			март			апрель				май				июнь						
				19	26	2	9	16	23	2	9	16	23	30	6	13	20	27	4	11	18			25	1	8
				25	1	8	15	22	1	8	15	22	29	5	12	19	26	3	10	17	24			31	7	14
1	Обязательные:																									
1.1	Посещение лекционных занятий	18	1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	18			
1.2	Защита лабораторных работ	5	8			+				+					+						+		40			
1.3	Выполнение курсовой работы	1	14																			+	14			
1.4	Промежуточное тестирование	1	8							+													8			
1.5	Итоговое тестирование	1	10																			+	10			
2	Творческий рейтинг:																									
2.1	Участие в студенческой конференции	1	10																			+	10			
3	Форма контроля																									
	Итого																						100	Диф.зач.		

- 1) При условии выполнения всех обязательных контрольных точек студент может получить 90 баллов, что соответствует оценке «отлично» (61-69 баллов – «удовлетворительно», 70-85 баллов – «хорошо», 86-100 баллов – «отлично»).
- 2) Для получения более высокой оценки студент может повысить количество баллов за счет участия в творческом рейтинге.
- 3) Для всех контрольных точек указано максимальное количество баллов.

