САМАРСКИЙ ПОЛИТЕХ
Опорный университет

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Самарский государственный технический университет»
(ФГБОУ ВО «СамГТУ»)
Филиал ФГБОУ ВО «СамГТУ» в г. Белебее Республики Башкортостан

УТВЕРЖДАЮ
Директор филиала ФГБОУ ВО «СамГТУ»
в г. Белебее Республики Башкортостан
_____ Л.М. Инаходова
26 мая 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.В.03.02 «Объектно-ориентированное программирование»

Код и направление подготовки (специальность)	**09.03.02 Информационные системы и технологии**
Направленность (профиль)	**Информационные системы и технологии**
Квалификация	**Бакалавр**
Форма обучения	**Заочная**
Год начала подготовки	**2022**
Выпускающая кафедра	**Инженерные технологии**
Кафедра-разработчик	**Инженерные технологии**
Объем дисциплины, ч. / з.е.	**180 / 5**
Форма контроля (промежуточная аттестация)	**Экзамен**

Белебей 2022 г.

Рабочая программа дисциплины (далее – РПД) разработана в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) 09.03.02 «Информационные системы и технологии», утвержденного приказом Министерства образования и науки РФ от 19 сентября 2017 г. № 926 , и соответствующего учебного плана.

Разработчик РПД:

доцент, к.т.н.		З.Ф. Камальдинова
(должность, степень, ученое звание)	(подпись)	(ФИО)

РПД рассмотрена и одобрена на заседании кафедры 26 мая 2022 г., протокол № 4.

Заведующий кафедрой	к.т.н., доцент	А.А. Цынаева
	(степень, ученое звание, подпись)	(ФИО)

СОГЛАСОВАНО:

Руководитель образовательной программы	доцент, к.т.н.	Е.Е. Ярославкина
	(степень, ученое звание, подпись)	(ФИО)

СОДЕРЖАНИЕ

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

Универсальные компетенции

Таблица 1

Наименование категории (группы) компетенций	Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
не предусмотрены учебным планом				

Общепрофессиональные компетенции

Таблица 2

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
не предусмотрены учебным планом			

Профессиональные компетенции

Таблица 3

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
ПК-2	Способность выполнять проектирование и графический дизайн интерактивных пользовательских интерфейсов	**ПК-2.3** Описывает и реализовывает логику работы элементов интерфейса пользователя, их взаимосвязи и взаимодействия с учетом возможностей целевых платформ	**У2 ПК-2.3** Уметь: Создавать интерактивные прототипы интерфейса **У2 ПК-2.3** Уметь: Создавать интерактивные прототипы интерфейса **В3 ПК-2.3** Владеть: Способностью описывать логику работы элементов интерфейса, их взаимосвязь, взаимодействие и варианты состояний
ПК-3	Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО	**ПК-3.1** Проектирует, разрабатывает, использует и документирует программные интерфейсы информационных систем	**З1 ПК-3.1** Знать: Методы и средства проектирования и документирования программных интерфейсов **В2 ПК-3.1** Владеть: Проектированием программных интерфейсов информационных систем
		ПК-3.2 Проектирует и реализовывает структуры, базы и хранилища данных	**У1 ПК-3.2** Уметь: Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов
		ПК-3.4 Использует типовые решения и библиотеки для реализации информационных систем с учетом особенностей архитектур различных целевых платформ	**З4 ПК-3.4** Знать: Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения **У3 ПК-3.4** Уметь: Использовать существующие типовые решения и шаблоны проектирования программного обеспечения
		ПК-3.5 Производит отладку, сборку и проверку работоспособности программного обеспечения	**З5 ПК-3.5** Знать: Основные принципы отладки и тестирования программных продуктов

2. Место дисциплины (модуля) в структуре образовательной программы

Место дисциплины в структуре образовательной программы: часть, формируемая участниками образовательных отношений.

Таблица 4

Код компетенции	Предшествующие дисциплины	Параллельно осваиваемые дисциплины	Последующие дисциплины
ПК-2		Производственная практика: технологическая (проектно-технологическая) практика; Проектирование человеко-машинного взаимодействия; Основы HTML, CSS и JS	Практико-ориентированный проект; Проектирование и разработка сетевых приложений; Надежность и оценка качества информационных систем; Эксплуатация информационных систем; Корпоративные информационные системы; Концептуальное проектирование и управление разработкой информационных систем; Проектирование и разработка интерфейсов информационных систем; Документирование информационных систем; Производственная практика: преддипломная практика
ПК-3	Офисное программирование и электронные форматы данных	Основы HTML, CSS и JS; Производственная практика: технологическая (проектно-технологическая) практика	Проектирование и разработка сетевых приложений; Проектирование баз и хранилищ данных; Практико-ориентированный проект; Концептуальное проектирование и управление разработкой информационных систем; Документирование информационных систем; Эксплуатация информационных систем; Проектирование и разработка интерфейсов информационных систем; Корпоративные информационные системы; Моделирование информационных процессов и систем ; Промышленная электроника и робототехника; Математические основы моделирования информационных систем; Производственная практика: преддипломная практика

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 обучающихся

Таблица 5

Вид учебной работы	Всего часов	Курс 2
Аудиторная контактная работа (всего), в том числе:	**18**	**18**
лекционные занятия (ЛЗ)	6	6
лабораторные работы (ЛР)	0	0
практические занятия (ПЗ)	12	12
Внеаудиторная контактная работа, КСР	**5**	**5**
Самостоятельная работа (всего), в том числе:	**148**	**148**

подготовка к практическим занятиям	74	74
самостоятельное изучение материала (конспект)	74	74
Формы текущего контроля успеваемости	Вопросы к устному опросу	Вопросы к устному опросу
Формы промежуточной аттестации	**экзамен**	**экзамен**
Контроль	**9**	**9**
ИТОГО: час.	**180**	**180**
ИТОГО: з.е.	**5**	**5**

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

Таблица 6

№ раздела	Наименование раздела дисциплины	Виды учебной нагрузки и их трудоемкость, часы						
		ЛЗ	ЛР	ПЗ	СРС	КСР	Конт-роль	Всего часов
1	Основы ООП	2	-	6	74	2	5	90
2	Программирование на Qt	4	-	6	74	3	4	90
Итого:		**6**	**0**	**12**	**148**	**5**	**9**	**180**

4.1. Содержание лекционных занятий

Таблица 7

№ ЛЗ	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Кол-во часов
Курс 2				
1	Основы ООП	Объектно-ориентированное программирование	Краткий обзор основных парадигм программирования. Основные принципы ООП. Абстрактные типы данных.	2
2	Программирование на Qt	Знакомство с Qt.	Обзор истории. Обзор настроек среды Qt Creator.	2
3	Программирование на Qt	Консольный проект Qt	Файлы проекта. Компиляция проекта. Вывод сообщений	2
			Итого за курс:	**6**
			Итого:	**6**

4.2. Содержание лабораторных занятий

Таблица 8

№ ЛР	Наименование раздела	Наименование лабораторной работы	Содержание лабораторной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Кол-во часов
не предусмотрены учебным планом				

4.3. Содержание практических занятий

Таблица 9

№ ПЗ	Наименование раздела	Тема практического занятия	Содержание практического занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Кол-во часов
Курс 2				
1	Основы ООП	Структуры (повторение)	Создание, заполнение, сохранение, обработка	2
2	Основы ООП	Классы и объекты	Создание класса, создание объектов класса.	2
3	Основы ООП	Класс, как область видимости	Доступ к полям объектов класса в области public.	2
4	Программирование на Qt	Компилятор gcc и интегрированная среда Qt Creator	Изучение основ работы с интегрированной средой разработки Qt Creator.	2
5	Программирование на Qt	Однофайловые проекты	Создание и сборка простейшего однофайлового проекта в командной строке	2
6	Программирование на Qt	Многофайловые проекты	Разработка проекта из нескольких файлов	2
			Итого за курс:	**12**
			Итого:	**12**

4.4. Содержание самостоятельной работы

Таблица 10

№ п/п	Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Кол-во часов
Курс 2				
1.	Основы ООП	подготовка к практическим	Производные классы и обработка исключительных	74

		занятиям	ситуаций	
	Программирование на Qt		Структура проекта. Основные типы Разработка приложений с графическим интерфейсом	
2.	Основы ООП	самостоятельное изучение материала (конспект)	Классы и методы	74
	Программирование на Qt		Структура проекта. Основные типы	
Итого за курс:				**148**
Итого:				**148**

5. Методические указания для обучающихся по освоению дисциплины (модуля)

1. Методические указания при работе на лекции

До лекции студент должен просмотреть учебно-методическую и научную литературу по теме лекции для того, чтобы иметь представление о проблемах, которые будут подняты в лекции.

Перед началом лекции обучающимся сообщается тема лекции, план, вопросы, подлежащие рассмотрению, доводятся основные литературные источники. Весь учебный материал, сообщаемый преподавателем, должен не просто прослушиваться. Он должен быть активно воспринят, т. е. услышан, осмыслен, понят, зафиксирован на бумаге и закреплен в памяти. Приступая к слушанию нового учебного материала, полезно мысленно установить его связь с ранее изученным. Следя за техникой чтения лекции (акцент на существенном, повышение тона, изменение ритма, пауза и т. п.), необходимо вслед за преподавателем уметь выделять основные категории, законы и определять их содержание, проблемы, предполагать их возможные решения, доказательства и выводы. Осуществляя такую работу, можно значительно облегчить себе понимание учебного материала, его конспектирование и дальнейшее изучение.

2. Методические указания при подготовке и работе на практическом занятии

Практические занятия по дисциплине проводятся в целях выработки практических умений и приобретения навыков в решении профессиональных задач.

Подготовка обучающегося к практическому занятию производится по вопросам, разработанным для каждой темы практических занятий и (или) лекций. В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы.

Работа студентов во время практического занятия осуществляется на основе заданий, которые выдаются обучающимся в начале или во время занятия. На практических занятиях приветствуется активное участие в обсуждении конкретных ситуаций, способность на основе полученных знаний находить наиболее эффективные решения поставленных проблем, уметь находить полезный дополнительный материал по тематике занятий. На практических занятиях обучающиеся должны уметь выработать определенные решения по обозначенной проблеме. В зависимости от сложности предлагаемых заданий, целей занятия, общей подготовки обучающихся преподаватель может подсказать обучающимся алгоритм решения или первое действие, или указать общее направление рассуждений. Полученные результаты обсуждаются с позиций их адекватности или эффективности в рассмотренной ситуации.

3. Методические указания по самостоятельной работе

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению с учетом потребностей и возможностей обучающегося.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала. Все новые понятия по изучаемой теме необходимо выучить наизусть.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий;
- на лекциях, практических занятиях;
- в контакте с преподавателем вне рамок расписания;
- на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т. д.;
- в методическом кабинете, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

4. Методические указания по подготовке к устному опросу

Самостоятельная работа обучающихся включает подготовку к устному опросу на семинарских занятиях. Для этого обучающийся изучает лекции, основную и дополнительную литературу, публикации, информацию из Интернет-ресурсов. Темы и вопросы к семинарским занятиям, вопросы для самоконтроля доводятся до обучающихся заранее. Эффективность подготовки обучающихся к устному опросу зависит от

качества ознакомления с рекомендованной литературой. Для подготовки к устному опросу необходимо ознакомиться с материалом по теме семинар и обратить внимание на усвоение основных понятий изучаемой темы, выявить неясные вопросы и подобрать дополнительную литературу для их освещения, составить тезисы выступления по отдельным проблемным аспектам. В среднем, подготовка к устному опросу по одному семинарскому занятию занимает от 2 до 4 часов

6. Перечень учебной литературы и учебно-методического обеспечения для самостоятельной работы

Таблица 11

№ п/п	Автор(ы), наименование, место, год издания (если есть, указать «гриф»)	Книжный фонд (КФ) или электрон. ресурс (ЭР)	Литература	
			учебная	для самост. работы
1.	Букунов С.В., Букунова О.В. Основы объектно-ориентированного программирования; Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||74339	ЭР	+	+
2.	Маляров А.Н. Объектно-ориентированное программирование; Самарский государственный технический университет, ЭБС АСВ, 2017.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||91772	ЭР	+	+
3.	Лисицин Д.В. Объектно-ориентированное программирование; Новосибирский государственный технический университет, 2010.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||44970	ЭР	+	+
4.	Николаев Е.И.Объектно-ориентированное программирование: учебное пособие; Северо-Кавказский федеральный университет, 2015.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||62967	ЭР	+	+
5.	Николаев Е.И. Объектно-ориентированное программирование: лабораторный практикум в 2-х частях. Часть 1; Северо-Кавказский федеральный университет, 2015.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||62966	ЭР	+	+
6.	Николаев Е.И. Объектно-ориентированное программирование: лабораторный практикум в 2-х частях. Часть 2; Северо-Кавказский федеральный университет, 2015.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||63218	ЭР	+	+
7.	Мейер Б. Объектно-ориентированное программирование и программная инженерия; Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||79706	ЭР		+
8.	Новиков П.В. Объектно-ориентированное программирование; Вузовское образование, 2017.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||64650	ЭР		+

Доступ обучающихся к ЭР НТБ СамГТУ (elib.samgtu.ru) осуществляется посредством электронной информационной образовательной среды университета и сайта НТБ СамГТУ по логину и паролю.

7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения

При проведении лекционных занятий используется мультимедийное оборудование. Организовано взаимодействие обучающегося и преподавателя с использованием электронной информационной образовательной среды университета.

Программное обеспечение

Таблица 12

№ п/п	Название	Способ распространения (лицензионное или свободно распространяемое)	Правообладатель (производитель)	Страна происхождения (иностранное или отечественное)
1.	LibreOffice Writer	свободно распространяемое	The Document Foundation	иностранное
2.	LibreOffice Impress	свободно распространяемое	The Document Foundation	иностранное
3.	LibreOffice Calc	свободно распространяемое	The Document Foundation	иностранное
4.	Adobe Reader	свободно распространяемое	Adobe Systems Incorporated	иностранное
5.	Справочно-правовая система «Консультант Плюс»	лицензионное	НПО «ВМИ»	отечественное
6.	Антивирус Касперского	лицензионное	Лаборатория Касперского	отечественное
7.	Яндекс.Браузер	свободно распространяемое	Яндекс	отечественное
8.	Архиватор 7-Zip	свободно распространяемое	7-zip.org	иностранное

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем

Таблица 13

№ п/п	Наименование	Краткое описание	Режим доступа
1	Электронно-библиотечная система IPRbooks	Электронно-библиотечная система	http://www.iprbookshop.ru/
2	Электронно-библиотечная система СамГТУ	Электронная библиотека СамГТУ	https://elib.samgtu.ru/
3	eLIBRARY.RU	Научная электронная библиотека	http://www.elibrary.ru/

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

Лекционные занятия

Аудитории для лекционных занятий укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории (наборы демонстрационного оборудования (проектор, экран, компьютер/ноутбук).

Практические занятия

Аудитории для практических занятий укомплектованы специализированной мебелью и техническими средствами обучения (проектор, экран, компьютер/ноутбук).

Самостоятельная работа

Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде СамГТУ:

- методический кабинет (ауд. 9).

10. Фонд оценочных средств по дисциплине

Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации представлен в Приложении 1.

Полный комплект контрольных заданий или иных материалов, необходимых для оценивания результатов обучения по дисциплине, практике хранится на кафедре-разработчике в бумажном и электронном виде.

Приложение 1

Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации

по дисциплине

Б1.В.03.02 «Объектно-ориентированное программирование»

Код и направление подготовки (специальность)	**09.03.02 Информационные системы и технологии**
Направленность (профиль)	**Информационные системы и технологии**
Квалификация	**бакалавр**
Форма обучения	**заочная**
Год начала подготовки	**2022**
Выпускающая кафедра	**Инженерные технологии**
Кафедра-разработчик	**Инженерные технологии**
Объем дисциплины, ч. / з.е.	**180 / 5**
Форма контроля (промежуточная аттестация)	**экзамен**

1. Перечень компетенций, индикаторов достижения компетенций и признаков проявления компетенций (дескрипторов), которыми должен овладеть обучающийся в ходе освоения образовательной программы

Универсальные компетенции

Таблица 1

Наименование категории (группы) компетенций	Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
не предусмотрены учебным планом				

Общепрофессиональные компетенции

Таблица 2

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
не предусмотрены учебным планом			

Профессиональные компетенции

Таблица 3

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
ПК-2	Способность выполнять проектирование и графический дизайн интерактивных пользовательских интерфейсов	**ПК-2.3** Описывает и реализовывает логику работы элементов интерфейса пользователя, их взаимосвязи и взаимодействия с учетом возможностей целевых платформ	**У2 ПК-2.3** Уметь: Создавать интерактивные прототипы интерфейса **У2 ПК-2.3** Уметь: Создавать интерактивные прототипы интерфейса **В3 ПК-2.3** Владеть: Способностью описывать логику работы элементов интерфейса, их взаимосвязь, взаимодействие и варианты состояний
ПК-3	Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО	**ПК-3.1** Проектирует, разрабатывает, использует и документирует программные интерфейсы информационных систем	**З1 ПК-3.1** Знать: Методы и средства проектирования и документирования программных интерфейсов **В2 ПК-3.1** Владеть: Проектированием программных интерфейсов информационных систем
		ПК-3.2 Проектирует и реализовывает структуры, базы и хранилища данных	**У1 ПК-3.2** Уметь: Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов
		ПК-3.4 Использует типовые решения и библиотеки для реализации информационных систем с учетом особенностей архитектур различных целевых платформ	**З4 ПК-3.4** Знать: Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения **У3 ПК-3.4** Уметь: Использовать существующие типовые решения и шаблоны проектирования программного обеспечения
		ПК-3.5 Производит отладку, сборку и проверку работоспособности программного обеспечения	**З5 ПК-3.5** Знать: Основные принципы отладки и тестирования программных продуктов

Матрица соответствия оценочных средств запланированным результатам обучения

Таблица 4

Код и индикатор достижения компетенции	Оценочные средства		
	Раздел 1	**Раздел 2.**	**Промежуточная аттестация**
	Основы ООП	Программирование на Qt	
	Вопросы к устному опросу		Вопросы экзаменационных билетов
ПК-2.3	У2 ПК-2.3 В3 ПК-2.3	У2 ПК-2.3 В3 ПК-2.3	У2 ПК-2.3 В3 ПК-2.3
ПК-3.1	З1 ПК-3.1 В2 ПК-3.1	З1 ПК-3.1 В2 ПК-3.1	З1 ПК-3.1 В2 ПК-3.1
ПК-3.2	У1 ПК-3.2	У1 ПК-3.2	У1 ПК-3.2
ПК-3.4	З4 ПК-3.4	У3 ПК-3.4	З4 ПК-3.4

			УЗ ПК-3.4
ПК-3.5	35 ПК-3.5	35 ПК-3.5	35 ПК-3.5

2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций в ходе освоения образовательной программы

2.1. Формы текущего контроля успеваемости

Текущий контроль успеваемости проводится в виде письменного/устного опроса и представляет собой ответы на вопросы.

Примерный перечень вопросов к устному опросу

Номер задания	Правильный ответ	Содержание вопроса	Компетенция	Время выполнения задания, мин
1.	В числе четырёх основных парадигм перечисляют процедурное программирование, структурное программирование, объектно-ориентированное программирование, функциональное программирование	Основные парадигмы программирования.	ПК-3	2
2.	Четыре основных принципа объектно-ориентированного программирования: абстракция, инкапсуляция, наследование, полиморфизм.	Основные принципы ООП.	ПК-3	2
3.	Абстрактный тип данных – это тип данных (набор значений и совокупность операций для этих значений), доступ к которому осуществляется только через интерфейс. Программу, которая использует АТД, называют клиентом, а программу, в которой содержится спецификация этого типа данных – реализацией.	Абстрактные типы данных.	ПК-3	2
4.	Область видимости представляет часть программы, в пределах которой можно использовать объект. Как правило, область видимости ограничивается блоком кода, который заключается в фигурные скобки. В зависимости от области видимости создаваемые объекты могут быть глобальными, локальными или автоматическими.	Что такое область видимости в C++?	ПК-3	2
5.	Объявление указывает уникальное имя сущности, а также сведения о ее типе и других характеристиках. В C++ точка, в которой объявляется имя, является точкой, в которой оно становится видимым для компилятора. Вы не можете ссылаться на функцию или класс, объявленный позднее в блоке компиляции.	Что такое объявление в C++?	ПК-3	2
6.	Конструктор по умолчанию (англ. default constructor), в объектно-ориентированных языках программирования — конструктор, который может быть вызван без аргументов.	Что такое конструктор по умолчанию C++?	ПК-3	2
7.	При каждом создании нового объекта класса вызывается конструктор класса.	Когда вызывается конструктор класса C ++?	ПК-3	2
8.	C++ позволяет указать несколько функций одного и того же имени в одной область. Эти функции называются перегруженными функциями или перегрузками. Перегруженные функции позволяют предоставлять различные семантики для функции в зависимости от типов и количества его аргументов.	Перегрузка функций.	ПК-3	
9.	Перегрузка операторов в программировании — один из способов реализации полиморфизма, заключающийся в возможности одновременного существования в одной области видимости нескольких различных вариантов применения оператора, имеющих одно и то же имя, но различающихся типами параметров, к которым они применяются.	Перегрузка операторов	ПК-3	2
10.	1. Выбираются только те перегруженные (одноименные) функции, для которых фактические параметры соответствуют формальным по количеству и типу (приводятся с помощью каких-либо преобразований). 2. Для каждого параметра функции (отдельно и по очереди) строится множество функций, оптимально отождествляемых по этому параметру (best matching). 3. Находится пересечение этих множеств: - если в нем содержится ровно одна функция – она и является искомой,	Алгоритм поиска оптимально отождествляемой функции для одного параметра.	ПК-3	2

	- если множество пусто или содержит более одной функции, генерируется сообщение об ошибке.			
11.	Наследование в C++ происходит между классами и имеет тип отношений «является». Класс, от которого наследуют, называется родительским (или «базовым», «суперклассом»), а класс, который наследует, называется дочерним (или «производным», «подклассом»).	Как работает наследование?	ПК-3	2
12.	Указатель — это переменная, в которой хранится адрес памяти объекта. Указатели широко используются как в C, так и в C++ для трех основных целей: для выделения новых объектов в куче, передача функций другим функциям	Что такое указатели в C ++?	ПК-3	2
13.	Открытое наследование означает «является». Все, что применимо к базовому классу, должно быть применимо также и производным от него, потому что каждый объект производного класса является также объектом базового класса. Имена в производных классах скрывают имена из базовых классов. При открытом наследовании это всегда нежелательно. Чтобы сделать скрытые имена видимыми, используйте using-объявления либо перенаправляющие функции.	Правила видимости при наследовании. Перекрытие имен.	ПК-3	2
14.	Указатели можно преобразовывать в ходе присваивания, инициализации, сравнения и выполнения других выражений.	Преобразования указателей.	ПК-3	2
15.	Виртуальный деструктор необходим, чтобы избежать возможной утечки ресурсов или другого неконтролируемого поведения объекта, в логику работы которого включен вызов деструктора.	Виртуальные деструкторы.	ПК-3	2
16.	Виртуальная функция — это функция-член, которую предполагается переопределить в производных классах. При ссылке на объект производного класса с помощью указателя или ссылки на базовый класс можно вызвать виртуальную функцию для этого объекта и выполнить версию функции производного класса	Что такое виртуальные функции в C++?	ПК-3	2
17.	Абстрактный класс определяет интерфейс для переопределения производными классами. Что такое чистые виртуальные функции (pure virtual functions)? Это функции, которые не имеют определения. Цель подобных функций - просто определить функционал без реализации, а реализацию определят производные классы.	Абстрактные классы. Чистые виртуальные функции.	ПК-3	2
18.	При работе в Qt Creator вы находитесь в одном из шести режимов: Начало, Правка, Отладка, Проекты, Справка и Вывод. Переключатель режимов позволяет вам быстро переключаться между задачами: редактирование, просмотра руководства по Qt Creator, настройка среды сборки и т.д. Начало - Отображает экран приветствия, позволяя вам быстро загружать недавние сессии или отдельные проекты. Этот режим вы увидите при запуске Qt Creator без указания ключей командной строки. Правка - Позволяет вам редактировать файлы проекта и исходных кодов. Боковая панель слева предоставляет различные виды для перемещения между файлами. Отладка - Предоставляет различные способы для просмотра состояния программы при отладке. Для описания как использовать этот режим смотрите Отладка с помощью Qt Creator. Проекты - Позволяет вам настроить как проекты должны быть собраны и запущены. Под списком проектов есть вкладки для настроек сборки, запуска и редактора. Справка - Показывает всю документацию, зарегистрированную в Qt Assistant, такую как документацию библиотеки Qt и Qt Creator. Вывод - Позволяет вам просматривать различные данные подробно, например, вывод сборки, а также вывод компилятора и программы. Эта информация также доступна в панелях вывода.	Обзор настроек среды Qt Creator.	ПК-2	2
19.	Библиотека Qt предоставляет набор контейнерных классов общего назначения, основанных на шаблонах. Эти классы могут использоваться для хранения элементов определенного типа. Контейнерные классы - классы с неявным совместным использованием данных, они реентерабельны, и они оптимизированы для быстрой работы, низкого потребления памяти и минимального увеличения кода (inline), результат в меньшем исполняемом файле. Кроме того, они потоко-безопасны в ситуациях, где они используются, как контейнеры только для чтения, всеми потоками, используемыми для доступа к ним.	Контейнерные классы в Qt	ПК-2	2
20.	Это - наиболее часто используемый контейнерный класс. Он хранит список значений заданного типа (T), доступ к которым осуществляется по индексу. Внутри, QList реализован используя	Список общего назначения QList.	ПК-2	2

	массив, гарантирующий быстрый доступ к элементам по индексу. Элементы могут быть добавлены в любой из двух концов списка используя QList::append() и QList::prepend(), или они могут быть вставлены в середину используя QList::insert(). QList наиболее оптимизированный, чем другие классы-контейнеры, чтобы развертываться в настолько маленький код насколько это возможно в исполняемом файле. QStringList унаследован от QList<QString>.			
21.	Предоставляет словарь (ассоциативное множество), который хранит соответствия ключей типа Key и значений типа T. QMap хранит данные, упорядоченные по ключу; если порядок не имеет значения, то класс QHash работает быстрее.	словарь (QMap)	ПК-2	2
22.	Имеет почти такой же API, как и QMap, но предоставляет значительно более быстрый поиск. QHash хранит свои данные в произвольном порядке.	QHash<Key, T>	ПК-2	2
23.	Виджеты — это «строительные блоки» для создания пользовательского интерфейса. Визуальные объекты на форме, такие как кнопки, метки, поля, меню, раскрывающиеся списки и т. д. Виджеты могут отображать данные и информацию о состоянии, получить ввод от пользователя и предоставлять контейнер для других виджетов, которые должны быть сгруппированы. Виджет, не встроенный в родительский виджет, называется окном.	Виджеты в Qt	ПК-2	2
24.	Компоновки - элегантный и гибкий способ для автоматического размещения дочерних виджетов внутри контейнера. Каждый виджет сообщает компоновщику свои требования к размеру посредством свойств sizeHint и sizePolicy, а компоновщик соответственно распределяет доступное пространство.	Компоновки	ПК-2	
25.	Каждый из виджетов, помещаемый на форму, должен быть размещен в нужном месте и с соответствующими размерами. Виджеты, размеры которых превышают размер формы, могут снабжаться полосами прокрутки, чтобы пользователь мог просмотреть все его содержимое. Qt предоставляет три основных способа управления размещением подчиненных виджетов на форме: абсолютное позиционирование, ручное управление размещением и менеджеры компоновки	Политика размеров	ПК-2	

2.2. Формы промежуточной аттестации

Промежуточная аттестация проводится в виде письменного/устного опроса, тестирования и представляет собой ответы на 2 вопроса и выполнение тестовых заданий.

Вопросы к экзамену

Номер задания	Правильный ответ	Содержание вопроса	Компетенция	Время выполнения задания, мин

1.	Класс - это шаблон или тип данных, который определяет состояние и поведение объектов, создаваемых на его основе. Класс описывает набор атрибутов (переменных) и методов (функций), которые могут быть использованы для работы с объектами этого класса.
	Определение класса состоит из ключевого слова class, за которым следует имя класса, и фигурных скобок, внутри которых указываются атрибуты и методы класса:
	
	cpp
	class MyClass {
	// атрибуты класса
	
	// методы класса
	};
	
	Атрибуты класса - это переменные, которые хранят состояние объектов класса. Они объявляются внутри класса и могут иметь различные типы данных:
		Классы и объекты. Определение класса.	ПК-3	2

	cpp
	class MyClass {
	int myInt; // атрибут класса типа int
	double myDouble; // атрибут класса типа double
	string myString; // атрибут класса типа string
	};
	Методы класса - это функции, которые могут выполнять операции с атрибутами класса или возвращать результаты. Они также объявляются внутри класса:
	cpp
	class MyClass {
	int myInt;
	
	void setInt(int value) { // метод класса для установки значения атрибута myInt
	myInt = value;
	}
	
	int getInt() { // метод класса для получения значения атрибута myInt
	return myInt;
	}
	};
	После определения класса можно создать объекты этого класса, используя оператор new:
	cpp
	MyClass obj; // создание объекта класса MyClass
	obj.setInt(10); // вызов метода setInt для установки значения атрибута myInt
	int value = obj.getInt(); // вызов метода getInt для получения значения атрибута myInt
	Классы позволяют организовывать код в более логически связанные блоки и упрощают работу с объектами, так как объединяют данные и функциональность в одном месте. Они являются основой объектно-ориентированного программирования.			

2.	Инкапсуляция в объектно-ориентированном программировании означает объединение данных и методов, работающих с этими данными, внутри класса. Это позволяет скрыть детали реализации и предоставить интерфейс для работы с объектами класса. Примеры использования инкапсуляции: 1. Класс "Автомобиль": Внутри класса хранятся данные об автомобиле, такие как марка, модель, год выпуска и т.д. Методы класса позволяют установить значения этих данных, получить их или выполнить операции с ними, например, запустить двигатель или остановить автомобиль. Внешний код может работать с объектами класса "Автомобиль" только через определенный интерфейс, скрывая детали реализации. 2. Класс "Банковский счет": Внутри класса хранятся данные о банковском счете, такие как номер счета, баланс и т.д. Методы класса позволяют установить значение баланса, выполнить операции по зачислению или списанию денег со счета и т.д. Внешний код может работать с объектами класса "Банковский счет" только через определенный интерфейс, скрывая детали реализации и обеспечивая безопасность операций с деньгами. 3. Класс "Студент": Внутри класса хранятся данные о студенте, такие как имя, возраст, средний балл и т.д. Методы класса позволяют установить или получить значения этих данных, а также выполнить операции, связанные со студентом, например, изменить его средний балл или добавить предмет в список изучаемых. Внешний код может работать с объектами класса "Студент" только через определенный интерфейс, скрывая детали реализации и обеспечивая безопасность данных. Инкапсуляция позволяет создавать более надежные и гибкие программы, так как изменения внутренней реализации класса не затрагивают внешний код, который использует объекты этого класса. Также она способствует повторному использованию кода и облегчает его поддержку и разработку.	Инкапсуляция - один из основных принципов ООП. Примеры.	ПК-3	2

3.	В C++ методы класса объявляются и определяются внутри определения класса. Объявление метода происходит в секции public, private или protected, в зависимости от доступности метода для других частей программы. Определение метода содержит его код, который выполняется при вызове метода.
	Синтаксис объявления метода класса в C++:
	cpp
	class MyClass {
	public:
	void myMethod(int arg1, int arg2);
	private:
		Методы класса. Объявление, определение и вызов методов.	ПК-3	2

	void myPrivateMethod();
	};
	Здесь myMethod - это имя метода, arg1 и arg2 - это аргументы метода. Метод myPrivateMethod объявлен как приватный и не доступен извне класса.
	Определение метода класса происходит за пределами определения класса. Оно содержит имя класса, к которому принадлежит метод, а также имя метода с указанием типов аргументов и возвращаемого значения (если есть).
	Синтаксис определения метода класса в C++:
	cpp
	void MyClass::myMethod(int arg1, int arg2) {
	// тело метода
	}
	
	void MyClass::myPrivateMethod() {
	// тело приватного метода
	}
	
	Вызов метода происходит через объект класса. Например, если у нас есть объект myObject класса MyClass, то вызов метода будет выглядеть так:
	cpp
	myObject.myMethod(arg1, arg2);
	При вызове метода значения аргументов передаются в соответствующие параметры метода.			

4.	Конструкторы и деструкторы - это специальные методы класса, которые выполняются при создании и удалении объекта соответственно. Конструкторы используются для инициализации объекта при его создании. Они имеют то же имя, что и класс, и не имеют возвращаемого значения. Конструктор может принимать аргументы, которые используются для инициализации полей объекта. Если конструктор не определен в классе, компилятор автоматически создает конструктор по умолчанию, который не принимает аргументов и не выполняет никаких действий.
	
	Пример объявления конструктора класса:
	
	cpp
	class MyClass {
	public:
	MyClass(); // конструктор по умолчанию
	MyClass(int arg); // конструктор с одним аргументом
	};
	Определение конструктора происходит за пределами определения класса. Оно содержит имя класса, к которому принадлежит конструктор, а также имя конструктора с указанием типов аргументов (если есть).
	
	cpp
	MyClass::MyClass() {
	// тело конструктора по умолчанию
	}
	
	MyClass::MyClass(int arg) {
	// тело конструктора с одним аргументом
	}
		Конструкторы класса.	ПК-3	2

5.	Конструкторы и деструкторы - это специальные методы класса, которые выполняются при создании и удалении объекта соответственно. Деструкторы используются для освобождения ресурсов, выделенных объекту, перед его удалением. Деструктор имеет то же имя, что и класс, с префиксом "~", и не принимает аргументов. Если деструктор не определен в классе, компилятор автоматически создает деструктор по умолчанию, который не выполняет никаких действий.
	
	Пример объявления деструктора класса:
	
	cpp
	class MyClass {
	public:
	~MyClass(); // деструктор
	};
	
	Определение деструктора происходит за пределами определения класса.
		Деструкторы класса. Перегрузка конструкторов.	ПК-3	2

	Оно содержит имя класса, к которому принадлежит деструктор.
	
	cpp
	MyClass::~MyClass() {
	  // тело деструктора
	}			

6.	Перегрузка конструкторов позволяет определить несколько конструкторов с разными списками аргументов. Компилятор выбирает подходящий конструктор на основе переданных аргументов при создании объекта.
	
	Пример перегрузки конструкторов:
	
	cpp
	class MyClass {
	public:
	  MyClass(); // конструктор по умолчанию
	  MyClass(int arg); // конструктор с одним аргументом
	  MyClass(int arg1, int arg2); // конструктор с двумя аргументами
	};
	
	MyClass::MyClass() {
	  // тело конструктора по умолчанию
	}
	
	MyClass::MyClass(int arg) {
	  // тело конструктора с одним аргументом
	}
	
	MyClass::MyClass(int arg1, int arg2) {
	  // тело конструктора с двумя аргументами
	}
	
	При создании объекта можно выбрать подходящий конструктор в зависимости от переданных аргументов:
	
	cpp
	MyClass obj1; // вызов конструктора по умолчанию
	MyClass obj2(10); // вызов конструктора с одним аргументом
	MyClass obj3(20, 30); // вызов конструктора с двумя аргументами
	В данном примере obj1 будет инициализирован с помощью конструктора по умолчанию, obj2 - с помощью конструктора с одним аргументом, а obj3 - с помощью конструктора с двумя аргументами.
		Перегрузка конструкторов.	ПК-3	2

7.	Конструктор копирования - это специальный метод класса, который создает новый объект, инициализируя его значениями другого объекта того же класса. Конструктор копирования вызывается при создании нового объекта с использованием существующего объекта в качестве аргумента.
	Пример объявления конструктора копирования класса:
	cpp
	class MyClass {
	public:
	  MyClass(); // конструктор по умолчанию
	  MyClass(const MyClass& other); // конструктор копирования
	};
	Определение конструктора копирования происходит за пределами определения класса. Оно содержит имя класса, к которому принадлежит конструктор копирования, а также имя конструктора с указанием типа аргумента (константная ссылка на объект класса).
	
	cpp
	MyClass::MyClass(const MyClass& other) {
	  // тело конструктора копирования
	}
	Конструктор копирования используется, когда необходимо создать копию существующего объекта. Например, при передаче объекта в функцию по значению или при инициализации нового объекта с использованием существующего.
	cpp
	MyClass obj1; // создание объекта obj1 с помощью конструктора по умолчанию
	MyClass obj2(obj1); // создание объекта obj2 с помощью конструктора копирования
		Конструктор копирования.	ПК-3	2

	В данном примере obj2 будет инициализирован с помощью конструктора копирования, используя значения объекта obj1.			
8.	Управление памятью с помощью операций new и delete позволяет программисту явно выделять и освобождать память во время выполнения программы. Преимущества использования операций new и delete: 1) Гибкость: операции new и delete позволяют динамически выделять и освобождать память, что позволяет программе адаптироваться к изменяющимся условиям и требованиям. 2) Эффективность: выделение памяти происходит только тогда, когда это необходимо, что позволяет эффективно использовать ресурсы компьютера. 3) Управление ресурсами: использование операций new и delete позволяет программисту контролировать использование памяти и избегать утечек ресурсов. 4) Динамическое создание объектов: операция new позволяет динамически создавать объекты во время выполнения программы, что может быть полезно при работе с различными типами данных или при создании объектов в зависимости от условий. 5) Изменение размера памяти: операция delete позволяет освободить память, выделенную под объект, что может быть полезно при изменении размера данных или при удалении объектов из контейнеров. В целом, использование операций new и delete обеспечивает гибкость, эффективность и контроль над использованием памяти в программе. Однако, необходимо быть внимательным при использовании этих операций, чтобы избежать утечек памяти или ошибок в управлении ресурсами.	Управление памятью с помощью операций new и delete, преимущества использования.	ПК-2	2

9.	Указатели на объекты и ссылки на переменные и методы класса через указатели - это механизмы, которые позволяют программисту работать с объектами и их членами (переменными и методами) с использованием указателей.
	Указатель на объект позволяет получить доступ к членам объекта (переменным и методам) и изменять их значения. Указатель на объект создается с использованием оператора "&" перед именем объекта. Например:
	Class obj;
	Class* ptr = &obj;
	
	В этом примере создается указатель ptr на объект obj класса Class. Ссылка на переменную или метод класса через указатель позволяет обращаться к этим элементам через указатель. Для доступа к элементам через указатель используется оператор "->". Например:
	
	int num = 10;
	int* ptr = #
	cout << *ptr; // Выводит значение переменной num
	
	В этом примере создается указатель ptr на переменную num, и затем через указатель выводится значение переменной num.
	Также можно создавать указатели на методы класса. Для этого используется тип указателя на метод, который имеет синтаксис тип_возвращаемого_значения (класс::*имя_указателя)(аргументы_метода). Например:
	
	class MyClass {
	public:
	void myMethod(int arg) {
	cout << "Метод MyClass::myMethod() вызван с аргументом " << arg << endl;
	}
	};
	
	int main() {
	MyClass obj;
	void (MyClass::*ptr)(int) = &MyClass::myMethod;
	(obj.*ptr)(10); // Вызов метода myMethod объекта obj через указатель ptr
	return 0;
	}
	
	В этом примере создается указатель ptr на метод myMethod класса MyClass. Затем этот метод вызывается через указатель ptr на объекте obj. Использование указателей на объекты и ссылок на переменные и методы
		Указатели на объекты и ссылки на переменные и методы класса через указатели.	ПК-	2

	класса через указатели позволяет программисту работать с объектами и их членами более гибко и динамически. Однако, необходимо быть внимательным при использовании указателей, чтобы избежать ошибок в работе с памятью и доступе к неверным адресам.			

10.	Виртуальные функции - это функции, определенные в базовом классе и переопределенные в производных классах. Они позволяют вызывать методы производного класса через указатель на базовый класс, что обеспечивает полиморфизм.
	Для доступа к обычным и виртуальным функциям через указатели используется оператор "->" или оператор "." в зависимости от того, является ли указатель на объект или ссылкой на объект.
	Например, если есть базовый класс Base и производный класс Derived, и у них есть виртуальная функция myFunction, то можно создать указатель на базовый класс и присвоить ему адрес объекта производного класса:
	
	Base* ptr = new Derived();
	
	Затем можно вызвать виртуальную функцию через указатель:
	
	ptr->myFunction();
	
	В этом случае будет вызвана переопределенная версия функции из производного класса Derived, даже если указатель имеет тип базового класса Base.
	
	Аналогично, если есть ссылка на объект:
	
	Base& ref = *ptr;
	ref.myFunction();
	
	Также будет вызвана переопределенная версия функции из производного класса Derived.
	Если же функция не является виртуальной, то будет вызвана версия функции из базового класса, даже если указатель или ссылка имеют тип производного класса.
		Виртуальные функции. Доступ к обычным и виртуальным функциям через указатели.	ПК-	2

11.	Для большинства приложений, алгоритм увеличения размера по умолчанию, предоставляемый Qt, выполняется так. Если вам требуется большая управляемость, то QVector<T>, QHash<Key, T>, QSet<T>, QString и QByteArray предоставляют три функции, которые позволяют контролировать и задавать столько памяти, сколько вам нужно для размещения элементов: capacity() возвращает количество элементов, для которых выделена память (для QHash и QSet - это количество участков памяти в хэш-таблице). reserve(size) явно предварительно выделяет память для size элементов. squeeze() освобождает любую память не требующуюся для хранения элементов. Если вы приблизительно знаете сколько элементов вы разместите в контейнере, то вы можете начать с вызова reserve(), а затем, заполнив контейнер, вы можете вызвать squeeze() чтобы освободить дополнительно выделенную память.	Стратегии увеличения размера	ПК-2	2
12.	В роли графического интерфейса выступают такие элементы: иконки; меню; списки; рисунки и схемы; другие графические элементы.	Какие элементы могут выступать в роли графического интерфейса?	ПК-2	2
13.	QObject является базовым классом для почти всех классов Qt. Исключением являются только классы, которые должны быть достаточно «лёгкими» (экземпляры которых должны занимать как можно меньше памяти) и классы, объекты которых должны копироваться (QObject не поддерживает копирования), а также контейнерные классы. Все виджеты Qt наследуют QObject (класс QWidget является потомком QObject). QObject реализует все базовые особенности, которыми обладают классы Qt: • мощный механизм взаимодействия между объектами с помощью сигнально-слотовых соединений; • иерархические взаимосвязи между объектами, позволяющие объединять их в объектные деревья; • управление памятью; • «умные» указатели, позволяющие отслеживать уничтожение объекта; • поддержка свойств; • таймеры;	Класс QObject.	ПК-2	2

	• обработка событий и фильтры событий; • метаинформация о типе объекта, его свойства и т. п			
14.	Управление памятью важно, поскольку неуправляемое выделение памяти приводит к утечке памяти в программе — ситуации, когда программа не освобождает память. Если такое неуправляемое выделение памяти повторяется периодически, а сама программа выполняется относительно долго, то со временем программа будет занимать всё больше места в оперативной памяти. Наконец, таким образом можно исчерпать всю доступную память, что приведёт к аварийному завершению программы и исчерпанию ресурсов операционной системы	Управление памятью	ПК-2	2
15.	Когда родительский объект удаляют, дочерние объекты тоже будут удалены из памяти перед удалением родительского. То же самое происходит и при удалении виджетов — все дочерние виджеты в визуальной иерархии тоже удаляются. Это позволяет управлять высвобождением памяти в программе	Иерархии объектов.	ПК-2	2
16.	В Qt существует механизм, который позволяет обрабатывать различные события, происходящие в системе и в самой программе. Визуальные элементы пользовательского интерфейса получают события от устройств ввода информации, которыми управляет пользователь: мышки, клавиатуры и т.п. Также события могут поступать от объектов изнутри приложения. Таким образом бывают: • события, возникшие спонтанно и поступающие от оконной системы, такие как нажатие клавиш, движения мышкой, манипуляции с окном программы (spontaneous events); • события, отправленные изнутри программы, созданные объектами в программе и направленные в цикл обработки событий (posted events) или напрямую к другому объекту (sent events).	События (Events). Обработка событий (Event handling).	ПК-2	2
17.	Диалог выбора файла для открытия и сохранения, диалог выбора шрифта, окна сообщений об ошибках являются примерами диалоговых окон, с которыми часто приходится сталкиваться при работе с программами. Такие диалоги обычно имеют стандартные для всех программ в системе вид и функциональность	Стандартные диалоги.	ПК-2	2
18.	Часто бывает необходимо добавить в программу дополнительные файлы такие как изображения и значки, используемые при оформлении интерфейса, звуковые файлы для уведомлений пользователя, сценарии, выполняемые программой и т.д. В таких случаях можно воспользоваться преимуществами, которые предоставляет ресурсная система Qt. Добавляя файл как ресурс в программу, мы указываем, что хотим включить данные, содержащиеся в этом файле в исполняемый файл. Таким образом скомпилированная программа будет содержать все ресурсные файлы внутри. Средства Qt позволяют обращаться к этим данным, и считывать файлы в ресурсах так же, как и обычные файлы в файловой системе.	Ресурсы программы.	ПК-2	2
19.	Алгоритм работы с дизайнером форм выглядит примерно так: 1. Примерно разместить виджеты на форме; 2. Применить к виджетам компоновки, начиная с наиболее вложенных виджетов; 3. Добавить меню, панели, настроить действия; 4. Настроить объекты на форме с помощью редактора свойств; 5. Сделать сигнально-слотовые соединения, если необходимо; 6. Реализовать логику работы в исходном коде программы.	Примерный алгоритм работы с дизайнером форм	ПК-2	2
20.	Иногда использование методов для доступа к защищённым элементам класса из внешней среды может оказаться неудобным. На такой случай предусмотрен специальный обходной путь. Чтобы класс мог предоставлять внешним функциям доступ к своей закрытой части, используется механизм объявления дружественных функций (friend) класса. Внутрь описания класса помещается прототип функции, перед которым ставится ключевое слово friend.	Дружественные функции. Основное назначение. Пример дружественной функции.	ПК-2	2
21.	Как уже упоминалось, перегрузка, т. е. возможность создавать функции (например, методы класса) с одинаковыми именами и разными наборами параметров, вызываемые в разных ситуациях для решения однотипных задач — это одно из ключевых проявлений полиморфизма в C++. Однако кроме перегрузки функций C++ позволяет проделывать то же самое с большинством стандартных операторов.	Перегрузка операторов	ПК-2	2

Примерный перечень тестовых заданий к промежуточной аттестации

Номер задания	Правильный ответ	Содержание вопроса	Компетенция	Время выполнения задания, мин
1.	Г	Повторному использованию способствует объектный механизм наследования классов. Наследование позволяет классу иметь? а) одного и только одного непосредственного потомка б) одного и только одного непосредственного родителя в) одного или более непосредственных родителей г) одного или более непосредственных потомков	ПК-3	2
2.	А	Повторному использованию способствует объектный механизм родовых классов. Универсализация позволяет классу иметь? а) несколько родовых параметров б) один и только один родовой параметр в) родовые параметры, задающие типы г) родовые параметры, задающие функции	ПК-3	2
3.	Б, В	Родовое порождение а) создает в результате новый тип б) разрешает использовать в качестве фактического параметра универсальный класс в) это процесс подстановки фактических параметров вместо формальных параметров универсального класса г) требует, чтобы фактические родовые параметры представляли типы.	ПК-3	2
4.	Б. В, Г	Отметьте истинные высказывания а) механизм ограниченной универсальности поддерживается механизмом наследования б) статическая типизация облегчает читаемость и повышает надежность программы в) единственные допустимые операции над сущностью, чей тип является формальным родовым параметром, это операции, применимые к любому типу г) универсализация требуется только в типизированном языке, гарантирующем статически проверяемую безопасность типов	ПК-2	2
5.	Г	Безопасность типов и повторное использование приводят к конфликту интересов. Этот конфликт а) неразрешим б) разрешается за счет того, что при введении универсализации запрещается статическая проверка типов в) разрешается за счет того, что при статической типизации запрещаются родовые параметры г) удается разрешить, сохранив статическую проверку типов и универсализацию	ПК-2	2
6.	Г	Наследование и универсализация а) это конкурирующие механизмы б) работа одного из этих механизмов требует присутствия другого механизма в) работа одного из этих механизмов исключает присутствие другого механизма г) это взаимно-дополняющие механизмы	ПК-2	2
7.	А, В, Г	Корректно использовать сущность x, чей тип задан формальным родовым параметром G, можно? а) всюду, где разрешается использовать сущность x конкретного типа б) в арифметических выражениях типа x + y, где оба операнда имеют тип G в) слева от оператора присваивания x:= y, где выражение y также имеет тип G г) справа от оператора присваивания y:= x , где сущность y также имеет тип G	ПК-2	2
8.	Б	Универсальный класс – это? а) синоним родового класса б) синоним параметризованного класса в) класс, позволяющий решить проблему универсализации г) класс с параметрами, заданными типами	ПК-2	2
9.	Г	Отметьте истинные высказывания а) в языках без статической проверки типов универсализация не нужна б) ограниченная универсальность предоставляет меньше возможностей в сравнении с общим случаем в) статическая проверка типов невозможна при введении универсальных классов с родовыми параметрами г) родовые классы могут использоваться для описания общих контейнерных структур данных, реализуемых независимо от типа элементов, которые они содержат	ПК-3	2

10.	Г	Эффективно реализованный механизм универсализации требует минимальных затрат а) времени выполнения б) памяти, требуемой для выполнения в) размера сгенерированного кода г) времени компиляции.	ПК-2	2

Примерная структура билета

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования «Самарский государственный технический университет» (ФГБОУ ВО «СамГТУ»)
Филиал ФГБОУ ВО «СамГТУ» в г. Белебее Республики Башкортостан

Кафедра «Инженерные технологии»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

По дисциплине (модулю): «Объектно-ориентированное программирование» Семестр 4

Направление 09.03.02 «Информационные системы и технологии»

1. Дружественные функции. Основное назначение. Пример дружественной функции.
2. Перегрузка операторов

Составил:
доцент _____ З.Ф. Камальдинова
(подпись)
«____» _____ ____ г.

Утверждаю:
Зав.кафедрой _____ А.А. Цынаева
(подпись)
«____» _____ ____ г.

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций

3.1. Характеристика процедуры текущей и промежуточной аттестации по дисциплине

Таблица 5

№ п/п	Наименование оценочного средства	Периодичность и способ проведения процедуры оценивания	Методы оценивания	Виды выставляемых оценок	Способ учета индивидуальных достижений, обучающихся
1.	Вопросы к устному опросу	систематически на всех видах занятий /письменно и устно	экспертный	По пятибалльной шкале	рабочая книжка преподавателя
3.	Промежуточная аттестация – вопросы экзаменационных билетов	по окончании изучения дисциплины/ устно и письменно	экспертный	По пятибалльной шкале	экзаменационная ведомость, зачетная книжка

3.2. Критерии и шкала оценивания результатов изучения дисциплины во время занятий (текущий контроль успеваемости)

Критерии оценки и шкала оценивания вопросов к устному опросу

Таблица 6

Шкала оценивания	Критерии оценки	Кол-во баллов
«Отлично»	Студент показывает полные и глубокие знания программного материала, логично и аргументировано отвечает на поставленный вопрос, а также дополнительные вопросы, показатели рейтинга (все предусмотренные РПД учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному).	(66-100) баллов
«Хорошо»	Студент показывает глубокие знания программного материала, грамотно его излагает, достаточно полно отвечает на поставленный вопрос и дополнительные вопросы, умело формулирует выводы, допуская незначительные погрешности, показатели рейтинга (все предусмотренные РПД учебные задания выполнены,	(36-65) баллов

	качество выполнения ни одного из них не оценено максимальным числом баллов).	
«Удовлетворительно»	Студент показывает достаточные, но неглубокие знания программного материала; при ответе не допускает грубых ошибок или противоречий, однако в формулировании ответа отсутствует должная связь между анализом, аргументацией и выводами, для получения правильного ответа требуется уточняющие вопросы, достигнуты минимальные или выше показатели рейтинговой оценки при наличии выполнения предусмотренных РПД учебных заданий	(16-35) баллов
«Неудовлетворительно»	Ответы на вопросы даны не верно	(0-15) баллов

Общие критерии и шкала оценивания результатов для допуска к промежуточной аттестации

Таблица 7

Наименование оценочного средства		Балльная шкала
1.	Вопросы к устному опросу	0-100 баллов
	Итого:	100 баллов

Максимальное количество баллов за семестр – 100. Обучающийся допускается к экзамену при условии 46 и более набранных за семестр баллов.

3.3 Критерии и шкала оценивания результатов изучения дисциплины на промежуточной аттестации

Основанием для определения оценки на экзамене служит уровень освоения обучающимися материала и формирования компетенций, предусмотренных программой учебной дисциплины.

Успеваемость определяется оценками: 5 «отлично»; 4 «хорошо»; 3 «удовлетворительно»; 2 «неудовлетворительно».

Оценку «отлично» получает обучающийся, освоивший компетенции дисциплины на всех этапах их формирования **на 85-100 %,** показавший всестороннее, систематическое и глубокое знание учебного материала, умение свободно выполнять задания, предусмотренные рабочей программой, усвоивший основную и ознакомленный с дополнительной литературой, рекомендованной программой. Как правило, оценка «отлично» выставляется обучающимся, усвоившим взаимосвязь основных положений учебной дисциплины, необходимых для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебного материала.

Оценку «хорошо» заслуживает обучающийся, освоивший компетенции дисциплины на всех этапах их формирования **на 71-84 %,** обнаруживший полное знание учебного материала, успешно выполняющий предусмотренные рабочей программой задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка «хорошо» выставляется обучающимся, продемонстрировавшим систематическое владение материалом дисциплины, способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности, но допустившим несущественные неточности в ответе.

Оценку «удовлетворительно» получает обучающийся, освоивший компетенции дисциплины на всех этапах их формирования **на 51-70 %,** обнаруживший знание основного учебного материала в объёме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющийся с выполнением заданий, предусмотренных рабочей программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка «удовлетворительно» выставляется обучающимся, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для устранения под руководством преподавателя допущенных недочетов.

Оценка «неудовлетворительно» выставляется обучающемуся, освоившему компетенции дисциплины на всех этапах их формирования менее чем **на 51%,** обнаружившему пробелы в знаниях основного учебного материала, допустившему принципиальные ошибки в выполнении предусмотренных рабочей программой заданий.

Шкала оценивания результатов

Таблица 8

Процентная шкала (при ее использовании)	**Оценка в системе «неудовлетворительно – удовлетворительно – хорошо – отлично»**
0-50%	Неудовлетворительно
51-70%	Удовлетворительно
71-84%	Хорошо
85-100%	Отлично

Приложение 2

УТВЕРЖДАЮ
Директор филиала ФГБОУ ВО «СамГТУ»
в г. Белебее Республики Башкортостан

_____ Л.М. Инаходова
«____» _____ 20__ г.

Дополнения и изменения к рабочей программе дисциплины (модуля)

Б1.В.03.02 «Объектно-ориентированное программирование»

по направлению подготовки (специальности) 09.03.02 «Информационные системы и технологии» по направленности (профилю) подготовки «Информационные системы и технологии»

на 20__/20__ учебный год

В рабочую программу вносятся следующие изменения:

1) .....;
2) .....;

Разработчик дополнений и изменений:

(должность, степень, ученое звание)	(подпись)	(ФИО)

Дополнения и изменения рассмотрены и одобрены на заседании кафедры «____» _____ 20___ г., протокол № ____.

Заведующий кафедрой

(степень, звание, подпись)	(ФИО)

Приложение 3

Аннотация рабочей программы дисциплины

Б1.В.03.02 «Объектно-ориентированное программирование»

Код и направление подготовки (специальность) **09.03.02 Информационные системы и технологии**

Направленность (профиль) **Информационные системы и технологии**

Квалификация **бакалавр**

Форма обучения **заочная**

Год начала подготовки **2022**

Выпускающая кафедра **Инженерные технологии**

Кафедра-разработчик **Инженерные технологии**

Объем дисциплины, ч. / з.е. **180 / 5**

Форма контроля (промежуточная аттестация) **экзамен**

Курс	Час. / з.е.	Лек. зан., час.	Лаб. зан., час.	Практич. зан., час.	КСР	СРС	Контроль	Форма контроля
4	180 / 5	6	-	12	5	148	9	экзамен
Итого	180 / 5	6	-	12	5	148	9	экзамен

Универсальные компетенции:	
не предусмотрены учебным планом	
Общепрофессиональные компетенции:	
не предусмотрены учебным планом	
Профессиональные компетенции:	
ПК-2	Способность выполнять проектирование и графический дизайн интерактивных пользовательских интерфейсов
ПК-2.3	Описывает и реализовывает логику работы элементов интерфейса пользователя, их взаимосвязи и взаимодействия с учетом возможностей целевых платформ
ПК-3	Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО
ПК-3.1	Проектирует, разрабатывает, использует и документирует программные интерфейсы информационных систем
ПК-3.2	Проектирует и реализовывает структуры, базы и хранилища данных
ПК-3.4	Использует типовые решения и библиотеки для реализации информационных систем с учетом особенностей архитектур различных целевых платформ
ПК-3.5	Производит отладку, сборку и проверку работоспособности программного обеспечения

Содержание дисциплины охватывает круг вопросов, связанных с объектно-ориентированным программированием. Классы и объекты C++. Конструкторы и деструкторы. Статический полиморфизм, одиночное наследование, динамический полиморфизм, механизм виртуальных функций. Консольный проект Qt, контейнерные классы, работа с файлами, создание графического интерфейса средствами Qt, разработка диалогов.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, практические занятия, самостоятельная работа студента.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости в форме вопросов к устному опросу и промежуточный контроль в форме экзамена.