! "# "$% %$ & ' (% $ #) * + %,"- !
' (% +.
/ 01
2 3 1 /

456

7 8
#9 % +$:

(%; 9 ;$ = ; #9
? * "$ < #); "$:

; ; +)#
?+ +>+ ,<

?: :+
/BC+E 66(!

3 1 1G

A "": $ & : ! &&+ % + D
? +& % + +">+: + ,@

: % +. : ! $ %9+H ": >+ "$+	FI;FJ;F6
+& % + : % : ! $ %9+H ": >+	+. "$+ & >+ , "+"$ &, + $ = !++
+& % + : % "$+	& >+ ,"+"$ &, + $ = !++ % '+(
& '#) +.	( ) .

9 $1 $ '#K6 F6

G+"$ " ! " % +. ') L : ! &&, +">+: + ,

1 ! &&# " "$ %+ ? @

"$; : : % B5;F ; ; - 9
? * "$ < #); "$: < (подпись лица) $ @ ?+ +>+ ,< &+ +.@
(%

1 ! && ' (" ++ 9 , 4 56
ABCD +E . 6F66 !< : $0696 4 NH6F6B

% #EO+L 9 L 4 56

;$; ;< BC;F ; ; +)#
?#); "$: < ?: :+" < ?+ +>+ ,<

$% $"$% ,L (FI1FJ;F6@?F

"$;: : % BC;F ; ; +9
? * "$ < #); $: < ?: :+" < $ @ ?+ +>+ ,< &+ +.@
(%

& "$+$ + 9$ + "$+$#$ 45 : & $ +) "9 L ' $

>;<9;$; BC;F ; ; E)
? * "$ < #); "$: < ?: :+" < $ @ ?+ +>+ ,< &+ +.@
(%

Аннотация

Дисциплина «Кроссплатформенное программирование» входит в образовательную программу высшего образования по направлению подготовки/ специальности 09.03.02 «Информационные системы и технологии» направленности «Информационные системы и технологии в бизнесе». Дисциплина реализуется кафедрой «№42».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ПК-3 «Способен разрабатывать программное обеспечение, выполнять интеграцию программных модулей и компонентов»

ПК-4 «Способен оценивать качество программного обеспечения, в том числе проведение тестирования и исследование результатов»

ПК-5 «Способен разрабатывать техническую документацию на продукцию в сфере информационных технологий, управления технической информацией»

Содержание дисциплины охватывает круг вопросов, связанных с разработкой программного обеспечения, способного работать под управлением нескольких операционных систем. Рассматриваются основные элементы многозадачных многопользовательских операционных систем, с которыми взаимодействует прикладное программное обеспечение. С использованием командной оболочки BASH и высокоуровневого языка программирования Python изучаются вопросы управления файлами, процессами, обработки текстовых данных. Уделяется внимание таким аспектам разработки программного обеспечения, как документирование, тестирование, распространение, а также интеграция программных модулей и межпроцессное взаимодействие.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа студента.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 5 зачетных единиц, 180 часов.

Язык обучения по дисциплине «русский».

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Получение студентами необходимых знаний и навыков в области разработки прикладного программного обеспечения для различных программных и аппаратных платформ.

1.2. Дисциплина входит в состав части, формируемой участниками образовательных отношений, образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Профессиональные компетенции	ПК-3 Способен разрабатывать программное обеспечение, выполнять интеграцию программных модулей и компонентов	ПК-3.З.1 знать методологии разработки, методы и средства проектирования программного обеспечения; методологии и технологии проектирования и использования баз данных; методы и средства проектирования программных интерфейсов, баз данных; методы и средства сборки модулей и компонент программного обеспечения ПК-3.В.1 владеть навыками оценки времени и трудоемкости реализации требований к программному обеспечению; разработки технических спецификаций на программные компоненты и их взаимодействия; проектирования структур данных, проектирования баз данных, программных интерфейсов; разработки процедур сборки модулей и компонент программного обеспечения; проверки работоспособности выпусков программного продукта
Профессиональные компетенции	ПК-4 Способен оценивать качество программного обеспечения, в том числе проведение тестирования и исследование результатов	ПК-4.З.1 знать теорию тестирования, техники тестирования; стандарты в области тестирования; базовые понятия качества программного продукта и качества процесса разработки программного обеспечения; метрики и риски тестирования; теорию критериев качества программного продукта и качества процесса разработки программного обеспечения ПК-4.У.1 уметь определять цели тестирования; разрабатывать требования к тестированию; выбирать и комбинировать техники тестирования; определять наиболее значимые критерии качества программного продукта

		ПК-4.В.1 владеть навыками разработки требований к тестированию на основе требований к системе; определения цели, объекта и видов тестирования; разработки последовательности проведения работ; оценки покрытия кода тестовыми случаями; анализа пропущенных дефектов и причины их пропуска; проведения анализа рисков и выработки плана по снижению рисков
Профессиональные компетенции	ПК-5 Способен разрабатывать техническую документацию на продукцию в сфере информационных технологий, управления технической информацией	ПК-5.У.1 уметь компоновать документ на основе заданных источников; подготавливать графические схемы; анализировать техническую документацию и научно-техническую литературу, извлекать сведения, необходимые для решения поставленной задачи; составлять обобщенные описания явлений, процессов, объектов управления; описывать бизнес-процессы с помощью графических нотаций; разрабатывать требования к техническому документу и к комплекту технической документации; составлять календарный план выполнения полученного задания; разрабатывать технические задания и спецификации требований; разрабатывать описание системной или программной архитектуры; разрабатывать руководства пользователя; анализировать целевую аудиторию комплекта технической документации; разрабатывать требования к техническому документу ПК-5.В.1 владеть навыками разработки концепции рекламного материала; составления текста рекламного материала, подготовки иллюстраций; разработки слайд-шоу; изучения целевой аудитории документа, выяснение ее задач, потребностей в информации, уровня подготовки; разработки концепции технической статьи, составления ее текста подготовки иллюстраций; изучения документируемой продукции с точки зрения всех целевых аудиторий и с учетом их информационных потребностей

2. Место дисциплины в структуре ОП

Дисциплина базируется на знаниях, ранее приобретенных студентами при изучении следующих дисциплин:

- Информатика;
- Основы программирования;
- Технологии программирования.

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

- Методы и средства проектирования информационных систем.

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам
		№6
1	2	3
Общая трудоемкость дисциплины, ЗЕ/(час)	5/ 180	5/ 180
	10	10
, всего час.,	20	20
лекции (Л), (час)	10	10
лабораторные работы (ЛР), (час)	10	10
экзамен, (час)	9	9
, всего (час)	151	151
: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз.	Экз.

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 6					
Раздел 1. Принципы создания кроссплатформенного ПО Тема 1.1. Виды программных и аппаратных платформ для прикладного программного обеспечения Тема 1.2. Технологии программирования, применяемые в кроссплатформенной разработке	1		0		20
Раздел 2. Работа с объектами UNIX-подобных операционных систем Тема 2.1. Файловая система в операционных системах семейства UNIX Тема 2.2. Создание процессов и манипуляции с ними	3		3		40

Тема 2.3. Взаимодействие прикладной программы с операционной системой Тема 2.4. Сходства и различия различных операционных систем по модели взаимодействия с прикладным программным обеспечением					
Раздел 3. Применение языка Python Тема 3.1. Особенности кроссплатформенной разработки на Python 3.2. Реализация механизмов ввода/вывода данных в консольном приложении. Работа с текстовыми данными	1		3		30
Раздел 4. Продуктизация программного обеспечения Тема 4.1. Распространение программ в виде пакетов Python и контейнеров Docker Тема 4.2. Создание документации Тема 4.3. Виды ручного и автоматического тестирования программ Тема 4.4 Интеграция программ на Python с программами на других языках	2		0		30
Раздел 5. Параллельное программирование Тема 5.1. Виды объектов синхронизации Тема 5.2. Библиотека asyncio Тема 5.3. Приложения с графическим пользовательским интерфейсом	3		4		31
Итого в семестре:	10		10		151
Итого:	10	0	10	0	151

4.2. Содержание разделов и тем лекционных занятий**.**

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	Принципы создания кроссплатформенного ПО Тема 1.1. Виды программных и аппаратных платформ для прикладного программного обеспечения. Основные различия между архитектурами программных и аппаратных платформ. Классификация операционных систем. Основные различия архитектур процессоров. Многообразие операционных систем семейства UNIX. Тема 1.2. Технологии программирования, применяемые в кроссплатформенной разработке. Виды кроссплатформенности. Уровни абстракции над особенностями операционной системы и аппаратного обеспечения. Кроссплатформенность на уровне исходного кода. Кроссплатформенность на уровне исполняемых модулей.
2	Работа с объектами UNIX-подобных операционных систем Тема 2.1. Файловая система в операционных системах семейства UNIX. Типы файлов. Атрибуты файлов и система разграничения доступа. Гибкие и жесткие ссылки. Механизм подсчета ссылок. Основные команды для манипуляций с файловой системой. Тема 2.2. Создание процессов и манипуляции с ними. Способы запуска командных сценариев. Соединение простых команд в конвейер. Тема 2.3. Взаимодействие прикладной программы с операционной системой. Строение простых и сложных команд BASH. Перечень потоков ввода-вывода и управление ими. Соединение простых команд в конвейер. Использование переменных окружения. Манипуляции с аргументами командной строки. Тема 2.4. Сходства и различия различных операционных систем по модели взаимодействия с прикладным программным обеспечением.
3	Применение языка Python Тема 3.1. Особенности кроссплатформенной разработки на Python. Сферы применения Python. Работа в интерактивном режиме. Использование Python для расширения возможностей командной строки. Тема 3.2. Реализация механизмов ввода/вывода данных в консольном приложении. Работа с текстовыми данными. Строки, контейнеры и работа с ними. Особенности типизации в Python. Виды стандартных контейнеров и синтаксис работы с ними. Методы обработки строк. Преобразование текста в различные кодировки. Особенности работы с Unicode.
4	Раздел 4. Продуктизация программного обеспечения Тема 4.1. Распространение программ в виде пакетов Python и контейнеров Docker. Виртаульные окружения python. Управление зависимостями сложных программ. Плюсы и минусы виртуализации и контейнеризации. Тема 4.2. Создание документации. Использование библиотек автоматического документирования на примере sphinx. Тема 4.3. Виды ручного и автоматического тестирования программ. Модульное тестирование. Тема 4.4 Интеграция программ на Python с программами на других языках. Оформление кода в виде разделяемых модулей.
5	Параллельное программирование Тема 5.1. Виды объектов синхронизации. Разрешение конфликтов совместного доступа к ресурсам. Процессы и потоки. Тема 5.2. Библиотека asyncio. Блокирующие и неблокирующие вызовы. Тема 5.3. Приложения с графическим пользовательским интерфейсом. Принципы создания пользовательского интерфейса. Возможности, предоставляемые библиотекой Qt, и ее интеграция с Python. Механизм сигналов и слотов. Принципиальные различия в устройстве графических подсистем Windows и UNIX.

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ № п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	№ раздела дисциплины
Учебным планом не предусмотрено				
Всего				

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 6				

1	Разработка командного сценария оболочки BASH	3	3	2
2	Разработка приложения для обработки текстовых данных	3	3	3
3	Разработка программы с асинхронными подзадачами	4	4	5
	Всего:	10	10	

4.5. Курсовое проектирование/ выполнение курсовой работы
Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся
Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 6, час
1	2	3
Изучение теоретического материала дисциплины (ТО)	100	100
Подготовка к текущему контролю успеваемости (ТКУ)	10	10
Домашнее задание (ДЗ)	15	15
Контрольные работы заочников (КРЗ)	16	16
Подготовка к промежуточной аттестации (ПА)	10	10
Всего:	151	151

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.
Таблица 8– Перечень печатных и электронных учебных изданий

Шифр	Библиографическая ссылка / URL адрес	Количество экземпляров в библиотеке (кроме электронных экземпляров)
004 А 94	Обработка данных средствами командной оболочки Bash и языка Python : учебно-методическое пособие / А. С. Афанасенко ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - СПб. : Изд-во ГУАП, 2019. - 55 с.	20
004.4 К 84	Методы программирования и прикладные алгоритмы [Текст] : учебное пособие в 3 ч. Ч. 1 / Е. А. Крук, А. А. Овчинников ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - СПб. : Изд-во ГУАП, 2014. - 178 с.	44
	Саммерфилд, М. Python на практике. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2014. — 338 с. — Режим доступа: http://e.lanbook.com/book/66480	
004 Л 85	Технологии параллельного программирования [Текст] : учебное пособие / С. А. Лупин, М. А. Посыпкин. - М. : ФОРУМ: ИНФРА-М, 2015. - 208 с.	20
	Догадин, Н.Б. Архитектура компьютера [Электронный ресурс] : учебное пособие. — Электрон. дан. — М.: "Лаборатория знаний" (ранее "БИНОМ. Лаборатория знаний"), 2015. — 274 с. http://e.lanbook.com/books/element.php?pl1_id=66281	
004.9 О-74	Информационно-сетевые технологии: монография/ Л. А. Осипов, С. А. Яковлев; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - СПб.: ГОУ ВПО "СПбГУАП", 2008. - 296с.	43
004.4 К 60	Ubuntu 10. Библия пользователя [Текст] / Д. Н. Колисниченко. - М. и др. : Диалектика, 2010. - 592 с.	15

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
http://www.proklondike.com/books/linux/granneman_Linux_commands_2010.html	С. Граннеман - Linux. Карманный справочник. Необходимый код и команды. 2010
http://wiki.python.su/	Wiki Портала Python-программистов

8. Перечень информационных технологий

8.1. **Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.**
Перечень используемого программного обеспечения представлен в таблице 10.
Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
1	BASH 4.1
2	Python 3.7
3	Oracle VM VirtualBox
4	JetBrains PyCharm IDE Community Edition

8.2. **Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине**
Перечень используемых информационно-справочных систем представлен в таблице 11.
Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	53-08
2	Вычислительная лаборатория	33-02

10.Оценочные средства для проведения промежуточной аттестации

10.1. **Состав оценочных средств для проведения промежуточной аттестации**

Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену;

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	– обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий.
«хорошо» «зачтено»	– обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий.
«удовлетворительно» «зачтено»	– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий.
«неудовлетворительно» «не зачтено»	– обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений.

10.3. **Типовые контрольные задания или иные материалы.**

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
1	Определение кроссплатформенности. Примеры кроссплатформенного ПО	ПК-3.З.1
2	Программная и аппаратная кроссплатформенность.	ПК-3.З.1
3	Кроссплатформенность на уровне исходного кода.	ПК-3.З.1
4	Кроссплатформенность на уровне исполняемых модулей. Виртуальные машины.	ПК-3.З.1
5	Типы файлов в UNIX, перечень и назначение их атрибутов, команды для управления атрибутами файлов	ПК-3.З.1
6	Конвейерное выполнение команд в оболочке sh/bash.	ПК-3.З.1
7	Приемы перенаправления стандартных потоков ввода-вывода для приложений.	ПК-3.З.1
8	Приемы разделения стандартного потока вывода и стандартного потока ошибок.	ПК-3.З.1
9	Асинхронный запуск приложений с помощью служб cron и screen (tmux).	ПК-3.З.1
10	Область видимости переменных в сценариях bash.	ПК-3.З.1
11	Временные и постоянные значения переменных окружения.	ПК-3.З.1
12	Виды аргументов функции на языке Python: формальные, позиционные и словарные аргументы.	ПК-3.З.1
13	Понятие итератора и итерируемого объекта.	ПК-3.З.1
14	Стандартные контейнеры в Python: кортежи, списки и словари.	ПК-3.З.1
15	Использование функции в качестве аргумента. Синтаксис лямбда-выражений.	ПК-3.З.1
16	Обработка исключений в языке Python.	ПК-3.З.1

17	Быстрое создание и преобразование контейнеров с помощью «генераторных» выражений.	ПК-3.3.1 ПК-3.В.1
18	Проблемы создания кроссплатформенного графического пользовательского интерфейса.	ПК-3.3.1
19	Перечень и назначение основных модулей библиотеки Qt.	ПК-3.3.1
20	Назначение docker-контейнеров и запуск прикладных приложений в них.	ПК-3.3.1
21	Базовые элементы пользовательского интерфейса, предоставляемые Qt.	ПК-3.3.1
22	Механизм сигналов и слотов для организации взаимодействия объектов в Qt.	ПК-3.3.1
23	Интернационализация приложений с помощью и без помощи Qt	ПК-3.3.1
24	Подготовка дистрибутивов кроссплатформенных приложений.	ПК-3.3.1
25	Технологии непрерывной интеграции программного обеспечения.	ПК-3.3.1
26	Практика безопасной работы с текстом в однобайтной и многобайтной кодировке	ПК-3.3.1 ПК-3.В.1
27	Учет различий файловых систем различных операционных систем в коде приложения на языке Python	ПК-3.3.1
28	Потенциальные трудности, связанные с переносом программного обеспечения на процессоры другой архитектуры.	ПК-3.3.1
29	Автоматизированное создание документации на ПО с помощью модуля sphinx	ПК-3.3.1 ПК-5.У.1
30	Виды тестирования ПО	ПК-4.3.1
31	Обоснование паттерна «Разработка через тестирование» (test driven development)	ПК-4.3.1
32	Создание модульных тестов в программах на языке Python	ПК-3.3.1 ПК-4.У.1
33	Пример организации автоматизированного тестирования с помощью модуля pytest	ПК-3.3.1 ПК-4.В.1
34	Пример создания скрипта для автоматического документирования программного модуля на языке python	ПК-5.В.1
35	Написать функцию, способную успешно пройти заданные модульные тесты	ПК-4.3.1 ПК-4.У.1 ПК-4.В.1
36	В приведенной команде запуска модульного тестирования найти и исправить ошибку	ПК-4.3.1 ПК-4.У.1 ПК-4.В.1
37	По приведенному коду программы на языке python определить результат вызова программы	ПК-3.В.1
38	Записать команду в оболочке bash для запуска скрипта на произвольном языке, файл которого доступен только для чтения	ПК-3.3.1 ПК-3.В.1
39	Изменить приведенный код программы с целью достижения кроссплатформенности (правильной работы в ОС Linux и Windows)	ПК-3.В.1
40	Найти ошибку в приведенном сценарии для автоматического документирования кода	ПК-5.В.1

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета
	Учебным планом не предусмотрено

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов
	Не предусмотрено

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
1	Процессы в UNIX-подобных ОС, атрибуты процессов, методы создания процессов, сходства и различия работы с процессами в различных операционных системах с точки зрения прикладного программирования.
2	Интеграция программных модулей языков C/C++ и Python. Создание python-интерфейса к динамической библиотеке. Вызов python-кода из программы на C++.
3	Построение конвейеров обработки больших текстовых данных в оболочке BASH (с примерами использования утилит awk, sed, sort).
4	Приемы перенаправления стандартных потоков ввода-вывода в оболочке BASH (включая разделение и объединение потоков).
5	Способы обмена данными между одновременно запущенными процессами.
6	Создание и настройка виртуального окружения для программ на python
7	Контейнеризация прикладного программного пакета с помощью ПО docker.
8	Модульное тестирование прикладных программ и его включение в процесс непрерывной интеграции (на примере системы контроля версий Github или Gitlab)
9	Общие принципы создания кроссплатформенного графического интерфейса с помощью библиотеки Qt. Достоинства и недостатки модели signal-slot. Пример самостоятельно разработанного графического интерфейса.
10	Возможности использования паттерна разработки «Модель-Представление-Контроллер» и ему подобных в кроссплатформенной разработке ПО.
11	Профилирование и отладка приложений на языке python. Особенности отладки асинхронных приложений.
12	Повышение производительности программ на python с помощью асинхронных задач.

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11.Методические указания для обучающихся по освоению дисциплины

Целью изучения дисциплины является получение студентами необходимых знаний, умений и навыков в области создания и распространения прикладных программ, ориентированных на несколько различных программных или аппаратных платформ и предназначенных для построения информационных систем и технологий.

11.1. Методические указания для обучающихся по освоению лекционного материала

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

- Раздел 1. Принципы создания кроссплатформенного ПО.
- Тема 1.1. Виды программных и аппаратных платформ для прикладного программного обеспечения.
- Тема 1.2. Технологии программирования, применяемые в кроссплатформенной разработке.
- Раздел 2. Работа с объектами UNIX-подобных операционных систем.
- Тема 2.1. Файловая система в операционных системах семейства UNIX.
- Тема 2.2. Создание процессов и манипуляции с ними.
- Тема 2.3. Взаимодействие прикладной программы с операционной системой.
- Тема 2.4. Сходства и различия различных операционных систем по модели. взаимодействия с прикладным программным обеспечением.
- Раздел 3. Применение языка Python.
- Тема 3.1. Особенности кроссплатформенной разработки на Python.
- Тема 3.2. Реализация механизмов ввода/вывода данных в консольном приложении. Работа с текстовыми данными.
- Раздел 4. Продуктизация программного обеспечения.
- Тема 4.1. Распространение программ в виде пакетов Python и контейнеров Docker.
- Тема 4.2. Создание документации.
- Тема 4.3. Виды ручного и автоматического тестирования программ.
- Тема 4.4 Интеграция программ на Python с программами на других языках.
- Раздел 5. Параллельное программирование.
- Тема 5.1. Виды объектов синхронизации.
- Тема 5.2. Библиотека asyncio.
- Тема 5.3. Приложения с графическим пользовательским интерфейсом.

Методические указания по освоению лекционного материала имеются в библиотеке ГУАП: Обработка данных средствами командной оболочки Bash и языка Python : учебно-методическое пособие / А. С. Афанасенко ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - СПб. : Изд-во ГУАП, 2019. - 55 с.

Эти материалы также имеются в электронном виде по адресу: https://pro.guap.ru/get-material/496dc89861a4a6d4d58c9c668eb1ce50.

11.2. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

Вариант задания по каждой лабораторной работе обучающийся получает в соответствии с номером в списке группы. Перед проведением лабораторной работы обучающемуся следует внимательно ознакомиться с методическими указаниями по ее выполнению. В соответствии с заданием обучающийся должен подготовить необходимые данные, получить от преподавателя допуск к выполнению лабораторной работы,

выполнить указанную последовательность действий, получить требуемые результаты, оформить и защитить отчет по лабораторной работе.

Структура и форма отчета о лабораторной работе

Отчет о лабораторной работе должен включать в себя: титульный лист, формулировку задания, теоретические положения, используемые при выполнении лабораторной работы, описание процесса выполнения лабораторной работы, иллюстративный материал (таблицы, графики, схемы), полученные результаты и выводы.

Требования к оформлению отчета о лабораторной работе

По каждой лабораторной работе выполняется отдельный отчет. Титульный лист оформляется в соответствии с шаблоном (образцом) приведенным на сайте ГУАП (www.guap.ru) в разделе «Сектор нормативной документации». Текстовые и графические материалы оформляются в соответствии с действующими ГОСТами и требованиями, приведенными на сайте ГУАП (www.guap.ru) в разделе «Сектор нормативной документации».

Методические указания по освоению лекционного материала имеются в библиотеке ГУАП: Обработка данных средствами командной оболочки Bash и языка Python : учебно-методическое пособие / А. С. Афанасенко ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - СПб. : Изд-во ГУАП, 2019. - 55 с.

Эти материалы также имеются в электронном виде по адресу: https://pro.guap.ru/get-material/496dc89861a4a6d4d58c9c668eb1ce50.

11.3. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа включает в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ для обучающихся по заочной форме обучения.

11.4. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Текущий контроль успеваемости обеспечивается проверочными работами, направленными на проверку степени освоения лекционного материала, также учитываются сроки сдачи лабораторных работ.

11.5. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

- экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Система оценок при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программы высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП».

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой