



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение
высшего образования

«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФГБОУ ВО «ИГУ»

Факультет бизнес-коммуникаций и информатики

Кафедра естественнонаучных дисциплин

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине Б1.В.05 Культура разработки программного обеспечения с открытым
исходным кодом

направление подготовки 09.03.03 Прикладная информатика

направленность (профиль) Прикладная информатика (разработка программного
обеспечения)

Одобрено
УМК факультета бизнес-коммуникаций
и информатики

Разработан в соответствии с ФГОС ВО

с учетом требований проф. стандарта

Председатель УМК

В.К. Карнаухова

ФИО, должность, ученая степень, звание

подпись, печать

Разработчики:

(подпись)

старший преподаватель

(занимаемая должность)

А.М. Веснин

(инициалы, фамилия)

Цель фонда оценочных средств. Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Б1.В.05 Культура разработки программного обеспечения с открытым исходным кодом». Перечень видов оценочных средств соответствует рабочей программе дисциплины.

Фонд оценочных средств включает контрольные материалы для проведения текущего контроля (в следующих формах: практическое задание, тест, проект) и промежуточной аттестации в форме вопросов и заданий к зачету и зачету с оценкой.

Структура и содержание заданий – задания разработаны в соответствии с рабочей программой дисциплины «Б1.В.05 Культура разработки программного обеспечения с открытым исходным кодом».

1. Паспорт фонда оценочных средств

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-2 Способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение	ПК-2.1	Знать технологии разработки программного обеспечения: методы, средства, процедуры и инструменты
	ПК-2.2	Уметь внедрять и адаптировать прикладное программное обеспечение
	ПК-2.3	Владеть навыками решения задач реализации и модификации ПО: планирования и оценки проекта по разработке ПО; анализа системных и программных требований; проектирования алгоритмов, структур данных и программных структур; кодирования с использованием различных языков программирования и разметки; рефакторинга ПО; тестирования и отладки программного кода; сопровождения
ПК-3 Способность осуществлять тестирование компонентов информационных систем и по заданным сценариям	ПК-3.1	Знать основные методы тестирования компонентов информационных систем
	ПК-3.2	Уметь пользоваться специальным программным обеспечением для автоматизированного тестирования (при необходимости) на уровне запуска готовых тестов
	ПК-3.3	Владеть навыками разработки тестовых сценариев компонентов информационных систем, проведения тестирования, исследования и анализа результатов

2. Показатели и критерии оценивания компетенций, шкалы оценивания

2.1. Показатели и критерии оценивания компетенций

№ п/п	Раздел, тема	Код индикатора компетенции	Наименование ОС	
			ТК	ПА
1	Основы Open Source, базовые определения и принципы устройства	Нет!	Нет!	Нет!
2	Обзор открытых лицензий и типов проектов для которых они подходят.	Нет!	Нет!	Нет!
3	Стадии жизни проекта и инструменты для их достижения.	ПК-2.1, ПК-3.1	Пз	Нет!
4	Работа над MVP. Этапы разработки и проверка гипотез.	ПК-2.3	Пз	Нет!
5	Система контроля версий git. Основные понятия. Работа с локальным репозиторием. Локальное использование.	ПК-2.2, ПК-2.3	Пз	Нет!
6	Основные сценарии использования git. Работа с удаленными репозиториями.	ПК-3.2	Пз	Нет!
7	Знакомство с репозиториями и платформами размещения исходного кода программ на примере Github.	ПК-3.1	Пз	Нет!
8	Стандарты составления документации к коду и приложению	Нет!	Нет!	Нет!
9	Работа в командной строке	ПК-2.3	Пз	Нет!
10	Знакомство с рабочим окружением. Системное окружение. Создание виртуального окружения venv для проекта.	ПК-3.2	Пз	Нет!
11	Менеджеры пакетов Python. Сборка проекта.	ПК-2.1, ПК-3.3	Пз	Нет!
12	Компиляция кода с использование открытых компиляторов gcc.	ПК-2.1, ПК-2.3, ПК-3.3	Пз	Нет!
13	Системы сборки и конфигурирования make.	ПК-2.2	Пз	Нет!
14	Среды разработки. Основные возможности	ПК-2.2	Пз	Нет!
15	Виртуализация и контейнеризация в чем тут разница на примере Docker. Создаем контейнер на базе Linux и настраиваем там рабочее окружение для проекта Python	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-3.3	Тест, Пз	Тест
16	Оформление кода. Виды стилей. Автоматические средства для форматирования.	ПК-2.1, ПК-3.2	Пз	Нет!
17	Создаем переиспользуемый код. Собираем библиотеку для использования в других проектах.	ПК-2.2, ПК-3.2	Пз	Нет!

№ п/п	Раздел, тема	Код индикатора компетенции	Наименование ОС	
			ТК	ПА
18	Учимся читать и использовать в своем проекте чужой код с Github.	Нет!	Пз	Нет!
19	Знакомимся с автотестами и культурой разработки кода test-driven-development	ПК-3.1, ПК-3.2, ПК-3.3	Пз	Нет!
20	Логирование работы приложения.	ПК-2.3, ПК-3.3	Пз	Нет!
21	Управление вызовом приложений - автоматизация процессов, посредством планировщика задач crontab и systemctl.	ПК-2.1, ПК-3.1	Пз	Нет!
22	Работа над проектом. Оформление	ПК-2.3, ПК-3.3	Пз	Нет!
23	Работа над проектом. Оформление документации по коду	ПК-2.2, ПК-3.3	Пз	Нет!
24	Доработка проекта	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.3, ПК-3.2	Тест, Проект	Тест

2.2. Критерии оценивания результатов обучения для текущего контроля успеваемости и промежуточной аттестации

Оценочное средство	Критерии оценивания	Шкала оценивания
Практическое задание	Задание выполнено верно. Выбран оптимальный путь решения. Присутствует развернутое описание алгоритма решения	Отлично
	Задание выполнено верно. Допущены негрубые логические ошибки при описании алгоритма решения. Отсутствуют пояснения к решению задания	Хорошо
	Ход решения задания верный, но допущены ошибки приведшие к неправильному ответу	Удовлетворительно
	В работе получен неверный ответ, связанный с грубыми ошибками допущенными в ходе решения, либо решение отсутствует полностью	Неудовлетворительно
Тест	Студентом даны правильные ответы на 91-100% заданий	Отлично
	Студентом даны правильные ответы на 81-90% заданий	Хорошо
	Студентом даны правильные ответы на 71-80% заданий	Удовлетворительно
	Студентом даны правильные ответы менее чем на 70% заданий	Неудовлетворительно

Оценочное средство	Критерии оценивания	Шкала оценивания
Проект	Проект студентом завершён в полном объеме. Для естественнонаучного проекта представлена работоспособная практическая часть, правильно выполнены и обоснованы необходимые расчёты. Реализация практической части проекта соответствует техническому заданию или заданию преподавателя. В проекте обозначена актуальность выбора темы с опорой на анализ предметной области. Студент способен сформулировать и обосновать практическую значимость своей работы. Подготовлена презентация результатов работы. Студент ориентируется во всех этапах разработки проекта, уверенно отвечает на вопросы аудитории. Способен аргументированно обосновать концепцию проекта и выбор инструментов для разработки проекта. Студент способен выделить достоинства и недостатки своей работы и предложить способы устранения недостатков	Отлично
	Проект студентом в целом завершён. Выполнены ключевые задачи. Для естественнонаучного проекта практическая часть в целом работоспособна, но есть мелкие неустранённые недостатки, необходимые расчеты в целом выполнены верно, но есть небольшие замечания. Реализация проекта в целом соответствует техническому заданию или заданию преподавателя. В проекте обозначена, но недостаточно обоснована актуальность темы. Практическая значимость работы просматривается, студент в целом может её сформулировать. Подготовлена презентация результатов работы. Студент ориентируется в этапах разработки проекта, но отвечает не на все вопросы аудитории. Способен обосновать выбор инструментов для реализации проекта. Студент способен выделить достоинства и недостатки своей работы, но не может предложить способы устранения последних	Хорошо
	Проект студентом завершён только в основных пунктах задания. Выполнены только ключевые задачи с недостатками. Для естественнонаучного проекта практическая часть работоспособна не вполне, есть существенные неустранённые недостатки, необходимые расчеты выполнены с ошибками. Реализация проекта частично соответствует техническому заданию или заданию преподавателя. В проекте обозначена, но не обоснована актуальность темы. Практическая значимость работы сформулирована слабо. Подготовлена презентация результатов работы. Студент слабо ориентируется в этапах разработки проекта, отвечает только на некоторые вопросы аудитории. Плохо обосновывает выбор инструментов для реализации проекта. Студент способен выделить достоинства и недостатки своей работы после серии наводящих вопросов, но не может предложить способы устранения недостатков	Удовлетворительно
	Проект студентом не завершён. Неполностью выполнены или не выполнены совсем ключевые задачи. Для естественнонаучного проекта практическая часть не работоспособна или не начата, есть существенные неустранённые недостатки, необходимые расчеты выполнены с грубыми ошибками. Реализация проекта не соответствует техническому заданию или заданию преподавателя. В проекте необозначена, и не обоснована актуальность темы. Практическая значимость работы не сформулирована. Плохо подготовлена презентация результатов работы. Студент почти не ориентируется в этапах разработки проекта, не отвечает на вопросы аудитории. Плохо обосновывается выбор инструментов для реализации проекта. Студент не способен выделить достоинства и недостатки своей работы даже после серии наводящих вопросов	Неудовлетворительно

2.3. Оценочные средства для текущего контроля (примеры)

2.3.1. Материалы для компьютерного тестирования обучающихся

Общие критерии оценивания

Процент правильных ответов	Оценка
91% – 100%	5 (отлично)

Процент правильных ответов	Оценка
81% – 90%	4 (хорошо)
71% – 80%	3 (удовлетворительно)
Менее 70%	2 (неудовлетворительно)

Соответствие вопросов теста индикаторам формируемых и оцениваемых компетенций

№ вопроса в тесте	Код индикатора компетенции
1	ПК-3.1
2	ПК-3.2
3	ПК-3.3
4	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-3.3
5	ПК-2.1
6	ПК-3.1
7	ПК-2.2
8	ПК-2.3

Ключ ответов

№ вопроса в тесте	Номер ответа (или ответ, или соответствие)
1	c
2	c
3	../src/foo.py ../tests/__init__.py ../tests/test_foo.py
4	установка git, настройка репозитория
5	pip install flask
6	c
7	b
8	d

Перечень тестовых вопросов

№ 1. Задание с единственным выбором. Выберите один правильный ответ.

Мы находимся в каталоге проекта, в котором есть файл Pipfile со следующим содержанием. [packages] numpy = "*" [dev-packages] pytest = "*" Какой из пакетов был установлен с помощью pipenv с флагом -d

- a. ни один, такого флага не существует.
- b. оба, установщик сам определяет куда отнести пакет, -d значит distribute, распределить
- c. pytest
- d. numpy

№ 2. Задание с единственным выбором. Выберите один правильный ответ.

В корневом каталоге проекта есть папка src и tests. В tests есть файлы test_foo.py и __init__.py, в src foo.py и foo.pyc. Мы находимся в корневом каталоге что отобразится на экране если в терминале Linux ввести команду "echo ls"?

- a. foo.py foo.pyc __init__.py test_foo.py

- b. src tests
- c. ls
- d. ./src/foo.py ./src/foo.pyс ./tests/__init__.py ./tests/test_foo.py

№ 3. Задание открытой формы. Введите ответ.

В корневом каталоге проекта есть папка src и tests. В tests есть файлы test_foo.py и __init__.py, в src foo.py и foo.pyс. Мы находимся в корневом каталоге что отобразится на экране если в терминале Linux ввести команду "cd ./src && ls ./*/*.py"?

№ 4. Задание открытой формы. Введите ответ.

Какие действия нужно выполнить перед тем как начать использовать в своем проекте Git

№ 5. Задание открытой формы. Введите ответ.

Напишите команду для менеджера пакетов pip, которая установит пакет flask

№ 6. Задание с единственным выбором. Выберите один правильный ответ.

С какой целью в проектах с использованием виртуального окружения venv используется файл "requirements.txt"

a. для хранения списка всех используемых в проекте пакетов с обязательным указанием версий

b. для хранения параметров и настроек виртуального окружения

c. для хранения списка всех используемых в проекте пакетов

d. для хранения истории вызовов команд менеджера пакетов venv

№ 7. Задание с единственным выбором. Выберите один правильный ответ.

Начиная с версии 3.5, в Python рекомендуется использовать модуль venv для создания виртуальных окружений. Какая команда создаст виртуальное окружение с названием env?

a. python venv env

b. python -m venv env

c. venv env

d. python -m create env

e. python -m env venv

№ 8. Задание с единственным выбором. Выберите один правильный ответ.

С какой целью используют Docker?

a. Для повышения степени защиты приложения от сетевых атак

b. Для разработки кроссплатформенных приложений на Python

c. Для локальной разработки приложений под Linux

d. Для быстрой разработки, тестирования и развертывания контейнерных приложений

e. Для разработки крупных монолитных приложений рассчитанных на высокую загрузку

2.3.2. Практические задания для оценки компетенции «ПК-2.1»

№ 1. Составление презентации по стеку использованных инструментов с указанием лицензии.

№ 2. Настраиваем полученную библиотеку как зависимость в первоначальном проекте, проверяем установку / сборку / прохождение тестов.

№ 3. Пробуем подход TDD, описываем задачу и пишем тесты и затем основную

функциональность.

№ 4. Пишем скрипт для анализа и визуализации логов - количество событий разного вида, количество событий по времени и прочее.

2.3.3. Практические задания для оценки компетенции «ПК-2.2»

№ 5. Демонстрация достижимости решения с учетом технологий доступных студентам, проверка на модельной задаче.

№ 6. Пишем службу для запуска нашего приложения `systemctl`.

№ 7. Разработаем персональный сайт визитку в котором отразим свои навыки разработчика и ссылки на открытые репозитории для ознакомления с опытом и качеством кода.

№ 8. Снабдить свой код комментариями для получения документации.

2.3.4. Практические задания для оценки компетенции «ПК-3.3»

№ 9. Демонстрация достижимости решения с учетом технологий доступных студентам, проверка на модельной задаче.

№ 10. Поиск лучших практик для использования `git`, подготовка презентации с описание проблемы и команд `git`, которые ее решают.

№ 11. Создаем `merge` запрос. Создаем `deploy`-токен и разворачиваем проект на сервере.

№ 12. Создадим рабочее окружение для своего проекта и настраиваем зависимости.

№ 13. Масштабируем приложение с помощью `docker`.

№ 14. Делаем рефакторинг своего приложения в соответствии со стандартами форматирования.

№ 15. Настраиваем полученную библиотеку как зависимость в первоначальном проекте, проверяем установку / сборку / прохождение тестов.

№ 16. Выбрать несколько проекта и объединить их в одном проекте, настроить рабочее окружение и опубликовать результат в своем репозитории на `GitHub`.

№ 17. Пробуем подход `TDD`, описываем задачу и пишем тесты и затем основную функциональность.

№ 18. Пишем скрипт для анализа и визуализации логов - количество событий разного вида, количество событий по времени и прочее.

№ 19. Пишем службу для запуска нашего приложения `systemctl`.

№ 20. Снабдить свой код комментариями для получения документации.

2.3.5. Практические задания для оценки компетенции «ПК-2.3»

№ 21. Поиск лучших практик для использования `git`, подготовка презентации с описание проблемы и команд `git`, которые ее решают.

2.3.6. Практические задания для оценки компетенции «ПК-3.2»

№ 22. Собираем проект в `Build Distribution (wheel)`.

№ 23. Сделаем автоматическую сборку с помощью `sphinx` для публикации в виде `pdf` или `html`.

№ 24. Настраиваем среду разработки.

№ 25. Пробуем подход `TDD`, описываем задачу и пишем тесты и затем основную функциональность.

№ 26. Пишем скрипт для анализа и визуализации логов - количество событий разного вида, количество событий по времени и прочее.

№ 27. Разработаем персональный сайт визитку в котором отразим свои навыки разработчика и ссылки на открытые репозитории для ознакомления с опытом и качеством кода.

№ 28. Снабдить свой код комментариями для получения документации.

2.3.7. Практические задания для оценки компетенции «ПК-3.1»

№ 29. Пробуем подход TDD, описываем задачу и пишем тесты и затем основную функциональность.

№ 30. Пишем скрипт для анализа и визуализации логов - количество событий разного вида, количество событий по времени и прочее.

№ 31. Снабдить свой код комментариями для получения документации.

2.3.8. Проекты для оценки компетенции «ПК-2.2»

№ 32. Интеграция полученных знаний и навыков в проект.

2.3.9. Проекты для оценки компетенции «ПК-2.3»

№ 33. Интеграция полученных знаний и навыков в проект.

2.3.10. Проекты для оценки компетенции «ПК-3.2»

№ 34. Интеграция полученных знаний и навыков в проект.

3. Промежуточная аттестация

3.1. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и опыта деятельности

Зачет является заключительным этапом процесса формирования компетенций обучающегося при изучении дисциплины и имеет целью проверку и оценку знаний обучающегося по теории, и применению полученных знаний, умений и навыков при решении практических задач.

Зачет проводится по расписанию, сформированному учебно-методическим управлением, в сроки, предусмотренные календарным учебным графиком. Зачет принимается преподавателем, ведущим лекционные занятия.

Зачет проводится только при предъявлении обучающимся зачетной книжки и при условии выполнения всех контрольных мероприятий, предусмотренных учебным планом и рабочей программой дисциплины. Обучающимся на зачете представляется право выбрать один из билетов. Время подготовки к ответу составляет 30 минут. По истечении установленного времени обучающийся должен ответить на вопросы билета. Результаты зачета оцениваются по четырехбалльной системе и заносятся в зачетно-экзаменационную ведомость и зачетную книжку. В зачетную книжку заносятся только положительные оценки. Подписанный преподавателем экземпляр ведомости сдаётся не позднее следующего дня в деканат.

В случае неявки обучающегося на зачет в зачетно-экзаменационную ведомость делается отметка «не явка». Обучающиеся, не прошедшие промежуточную аттестацию по дисциплине, должны ликвидировать академическую задолженность в установленном локальными нормативными актами порядке.

3.2. Вопросы к зачету

№	Вопрос	Код компетенции
1.	С какой целью разработали лицензии для исходного кода и ПО (GPL, MIT, Apache, BSD, GNU GPL)?	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-3.3
2.	Что означает открытый (публичный, public) репозиторий и чем он отличается от закрытого (private) репозитория?	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-3.3
3.	Какую роль в IT сыграла история появления ОС Linux?	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-3.3
4.	Какие программы позволяют большим командам программистов, находящихся по всему миру, совместно работать над проектом?	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-3.3
5.	Что такое «Open Source»?	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-3.3

3.3. Вопросы к зачету с оценкой

№	Вопрос	Код компетенции
1.	Защита итогового проекта	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-3.3

3.4. Тематика курсовых работ

По данной дисциплине выполнение курсовых проектов (работ) не предусматривается.

3.5. Материалы для компьютерного тестирования обучающихся

Общие критерии оценивания

Процент правильных ответов	Оценка
91% – 100%	5 (отлично)
81% – 90%	4 (хорошо)
71% – 80%	3 (удовлетворительно)
Менее 70%	2 (неудовлетворительно)

Соответствие вопросов теста индикаторам формируемых и оцениваемых компетенций

№ вопроса в тесте	Код индикатора компетенции
1	ПК-3.1
2	ПК-3.2
3	ПК-3.3
4	ПК-2.1, ПК-2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-3.3
5	ПК-2.1
6	ПК-3.1
7	ПК-2.2
8	ПК-2.3

Ключ ответов

№ вопроса в тесте	Номер ответа (или ответ, или соответствие)
1	c
2	c
3	../src/foo.py ../tests/__init__.py ../tests/test_foo.py
4	установка git, настройка репозитория
5	pip install flask
6	c
7	b
8	d

Перечень тестовых вопросов

№ 1. Задание с единичным выбором. Выберите один правильный ответ.

Мы находимся в каталоге проекта, в котором есть файл Pipfile со следующим содержанием. [packages] numpy = "*" [dev-packages] pytest = "*" Какой из пакетов был установлен с помощью pipenv с флагом -d

- a. ни один, такого флага не существует.
- b. оба, установщик сам определяет куда отнести пакет, -d значит distribute, распределить
- c. pytest
- d. numpy

№ 2. Задание с единичным выбором. Выберите один правильный ответ.

В корневом каталоге проекта есть папка src и tests. В tests есть файлы test_foo.py и __init__.py, в src foo.py и foo.pyc. Мы находимся в корневом каталоге что отобразится на экране если в терминале Linux ввести команду "echo ls"?

- a. foo.py foo.pyc __init__.py test_foo.py
- b. src tests
- c. ls
- d. ./src/foo.py ./src/foo.pyc ./tests/__init__.py ./tests/test_foo.py

№ 3. Задание открытой формы. Введите ответ.

В корневом каталоге проекта есть папка src и tests. В tests есть файлы test_foo.py и __init__.py, в src foo.py и foo.pyc. Мы находимся в корневом каталоге что отобразится на экране если в терминале Linux ввести команду "cd ./src && ls ../**/*.py"?

№ 4. Задание открытой формы. Введите ответ.

Какие действия нужно выполнить перед тем как начать использовать в своем проекте Git

№ 5. Задание открытой формы. Введите ответ.

Напишите команду для менеджера пакетов pip, которая установит пакет flask

№ 6. Задание с единичным выбором. Выберите один правильный ответ.

С какой целью в проектах с использованием виртуального окружения venv используется файл "requirements.txt"

- a. для хранения списка всех используемых в проекте пакетов с обязательным указанием версий

- b. для хранения параметров и настроек виртуального окружения
- c. для хранения списка всех используемых в проекте пакетов
- d. для хранения истории вызовов команд менеджера пакетов `venv`

№ 7. Задание с единственным выбором. Выберите один правильный ответ.

Начиная с версии 3.5, в Python рекомендуется использовать модуль `venv` для создания виртуальных окружений. Какая команда создаст виртуальное окружение с названием `env`?

- a. `python venv env`
- b. `python -m venv env`
- c. `venv env`
- d. `python -m create env`
- e. `python -m env venv`

№ 8. Задание с единственным выбором. Выберите один правильный ответ.

С какой целью используют Docker?

- a. Для повышения степени защиты приложения от сетевых атак
- b. Для разработки кроссплатформенных приложений на Python
- c. Для локальной разработки приложений под Linux
- d. Для быстрой разработки, тестирования и развертывания контейнерных приложений
- e. Для разработки крупных монолитных приложений рассчитанных на высокую загрузенность