Министерство образования Калининградской области

государственное бюджетное учреждение Калининградской области

профессиональная образовательная организация

«Колледж информационных технологий и строительства»

(ГБУ КО ПОО «КИТиС»)

**Отчет по учебной практике**

УП.04 Сопровождение и обслуживание программного обеспечения компьютерных систем

по ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем

Специальность 09.02.07 «Информационные системы и программирование»

Сроки прохождения практики:

с «13» февраля 2023 г. по «4» марта 202\_ г.

Место практики ГБУ КО ПОО «КИТиС»

|  |  |
| --- | --- |
| Выполнил: | студент 2 курса,  группы ИСп 21-2к  Конопляник Дмитрий Андреевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград, 2023

[Основная часть 5](#_Toc128913087)

[1 Обеспечение внедрения и поддержки программного обеспечения компьютерных систем 5](#_Toc128913088)

[1.1 Инструктаж по технике безопасности 5](#_Toc128913089)

[1.2 Описание выбранного аппаратного и программного обеспечения 6](#_Toc128913090)

[1.3 Методы внедрения и анализа функционирования программного обеспечения 6](#_Toc128913091)

[1.4 Сценарий и техническое задание на внедрение ПО 7](#_Toc128913092)

[1.5 Загрузка и установка программного обеспечения 7](#_Toc128913093)

[1.6 Настройка отдельных компонентов программного обеспечения компьютерных систем и конфигурирование 10](#_Toc128913094)

[2 Проблемы совместимости и методы их устранения 10](#_Toc128913095)

[2.1 Тестирование на совместимость в безопасном режиме 10](#_Toc128913096)

[2.2 Инструменты повышения производительности программного обеспечения 10](#_Toc128913097)

[2.3 Средства диагностики оборудования 11](#_Toc128913098)

[2.4 Измерение и анализ эксплуатационных характеристик качества программного обеспечения 12](#_Toc128913099)

[2.5 Оптимизация и модификация ПО 12](#_Toc128913100)

[2.6 Разработка руководства оператора 12](#_Toc128913101)

[3 Этапы сопровождения и обслуживания ПО 13](#_Toc128913102)

[3.1 Разработка технического задания на сопровождение 13](#_Toc128913103)

[3.2 Обеспечение качества компьютерных систем в процессе эксплуатации 14](#_Toc128913104)

[3.3 Разработка сопровождающей документации 15](#_Toc128913105)

[4 Методы и средства защиты компьютерных систем 16](#_Toc128913106)

[4.1 Тестирование на ошибки ввода/вывода данных 16](#_Toc128913107)

[4.2 Тестирование интерфейса 16](#_Toc128913108)

[4.3 Тестирование контента 17](#_Toc128913109)

[4.4 Анализ рисков. Выявление первичных и вторичных ошибок 17](#_Toc128913110)

[4.5 Обнаружение вируса и устранение последствий его влияния 18](#_Toc128913111)

[4.6 Тестирование защиты программного обеспечения 18](#_Toc128913112)

[4.7 Работа с реестром. Работа с программой восстановления файлов и очистки дисков 18](#_Toc128913113)

[5 Индивидуальное задание 20](#_Toc128913114)

[5.1 Практические работы 20](#_Toc128913115)

[5.2 Проектирование проведения электронных выставок и покупательских конференций 26](#_Toc128913116)

[5.3 Характеристики программ с точки зрения влияния на их защищенность и результаты работы 26](#_Toc128913117)

[Заключение 28](#_Toc128913118)

Введение

PyCharm — это интегрированная среда разработки на языке Python, созданная компанией JetBrains. Среди людей, использующих Python эта IDE вторая по популярности после редактора кода Visual Studio Code.

В PyCharm есть все инструменты, чтобы **писать, отлаживать и тестировать код**. Также IDE позволяет использовать в проектах другие языки программирования, синхронизировать код с системами контроля версий и развёртывать его.

Базовый набор функций PyCharm можно расширять с помощью **плагинов,** которые позволяют, например, настраивать внешний вид интерфейса и подключать дополнительные инструменты.

PyCharm представляет собой большой набор инструментов:

* встроенный отладчик и инструмент запуска тестов
* профилировщик Python
* полнофункциональный встроенный терминал
* инструменты для работы с базами данных.

Основная часть

1 Обеспечение внедрения и поддержки программного обеспечения компьютерных систем

* 1. Инструктаж по технике безопасности

Перед работой:

* + Необходимо подготовить рабочее место.
  + Отрегулировать освещение и убедиться в отсутствии бликов на экране.
  + Проверить правильность подключения оборудования к электросети.
  + Проверить исправность проводов питания и отсутствие оголенных участков проводов.
  + Убедиться в наличии заземления системного блока, монитора и защитного экрана.
  + Протереть антистатической салфеткой поверхность экрана монитора и защитного экрана.
* Во время работы:

Работнику при работе на ПК запрещается:

- прикасаться к задней панели системного блока (процессора) при включенном питании;

- переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;

- допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и других устройств;

- производить самостоятельное вскрытие и ремонт оборудования;

- работать на компьютере при снятых кожухах;

- отключать оборудование от электросети и выдергивать электровилку, держась за шнур.

* + Продолжительность непрерывной работы с компьютером без регламентированного перерыва не должна превышать 2-х часов.
  + Во время регламентированных перерывов с целью снижения нервно - эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения развития познотонического утомления выполнять комплексы упражнений.
* Во время аварийных ситуаций:
  + Во всех случаях обрыва проводов питания, неисправности заземления и других повреждений, появления гари, немедленно отключить питание и сообщить об аварийной ситуации руководителю.
  + Не приступать к работе до устранения неисправностей.
  + При получении травм или внезапном заболевании немедленно известить своего руководителя, организовать первую доврачебную помощь или вызвать скорую медицинскую помощь.
* По окончанию работы:
* Отключить питание компьютера.
* Привести в порядок рабочее место.
* Выполнить упражнения для глаз и пальцев рук на расслабление.
  1. Описание выбранного аппаратного и программного обеспечения

PyCharm — это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для написания кода и визуальный отладчик.

* 1. Методы внедрения и анализа функционирования программного обеспечения

Внедрение проводится втечение 10 минут, а далее весь анализ функционирования производится на вычислительной технике пользователя и до того момента, когда ПО не будет находиться на стадии внедрения заказчику.

* 1. Сценарий и техническое задание на внедрение ПО

Сценарии внедрения - это сводка задач внедрения продукта.

Внедрение программного обеспечения — процесс настройки программного обеспечения под определённые условия использования, а также обучения пользователей работе с программным продуктом.

Сценарий внедрения PyCharm:

* Проверить соответствуют ли характеристики ПК требованиям
* Скачать и установить нужную версию
* Проверить нет ли ошибок или неисправностей после установки
* Настроить программу
  1. Загрузка и установка программного обеспечения
* Загрузите установщик .exe с веб-страницы приложения Toolbox.
* Запустите установщик и следуйте инструкциям мастера.
* После запуска приложения Toolbox щелкните его значок  в области уведомлений и выберите продукт, который вы хотите установить.
* Чтобы установить определенную версию, щелкните  и выберите Доступные версии.

Рисунок 1 – иконка приложения

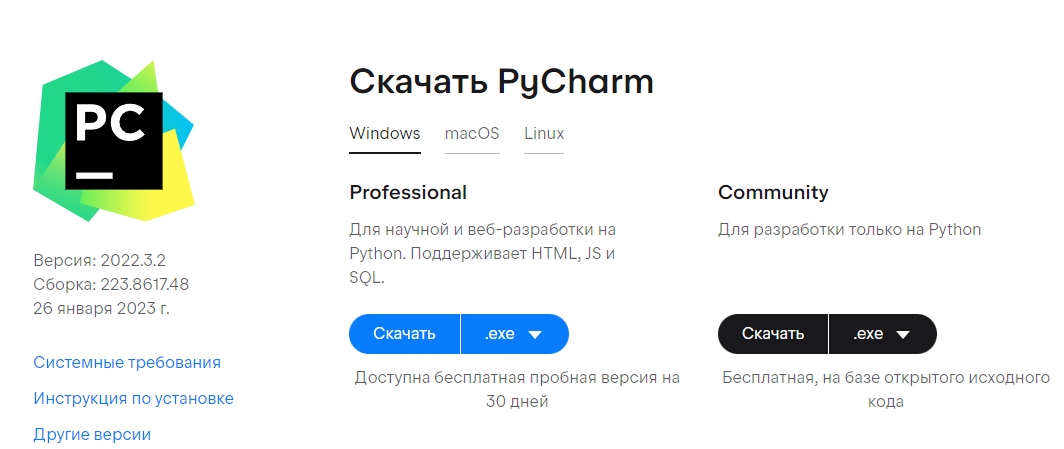


Рисунок 2 – выбор версии ПО для установки

* Загрузите установщик.
* Запустите установщик и следуйте инструкциям.

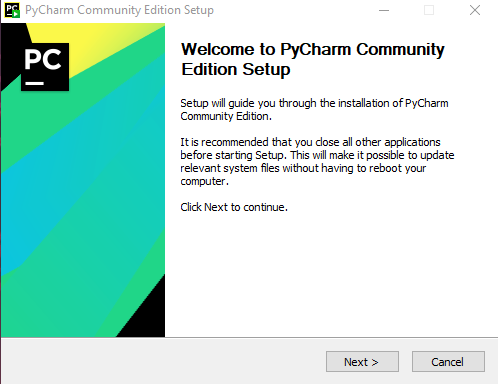


Рисунок 3 – установщик PyCharm

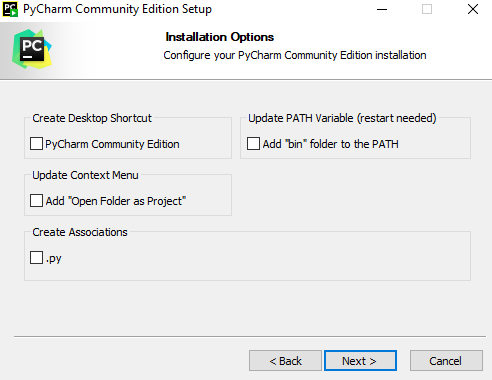


Рисунок 4 – выбор опций установки

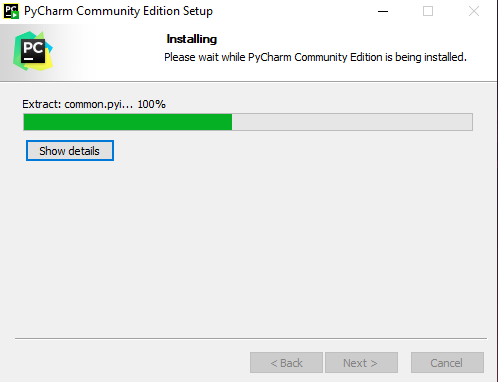


Рисунок 5 – процесс установки

* 1. Настройка отдельных компонентов программного обеспечения компьютерных систем и конфигурирование

Настройка производительности невозможна

Настройка отдельных компонентов необязательна. Так что особой настройки не требует.

В настройках можно указать путь, по которому находятся пользовательские настройки, а также можно указать свой путь к виртуальной машине.

1. Проблемы совместимости и методы их устранения
   1. Тестирование на совместимость в безопасном режиме

* Возможность возникновения проблем с эмуляцией
* Могут возникнуть проблемы с интерпретатором
  1. Инструменты повышения производительности программного обеспечения

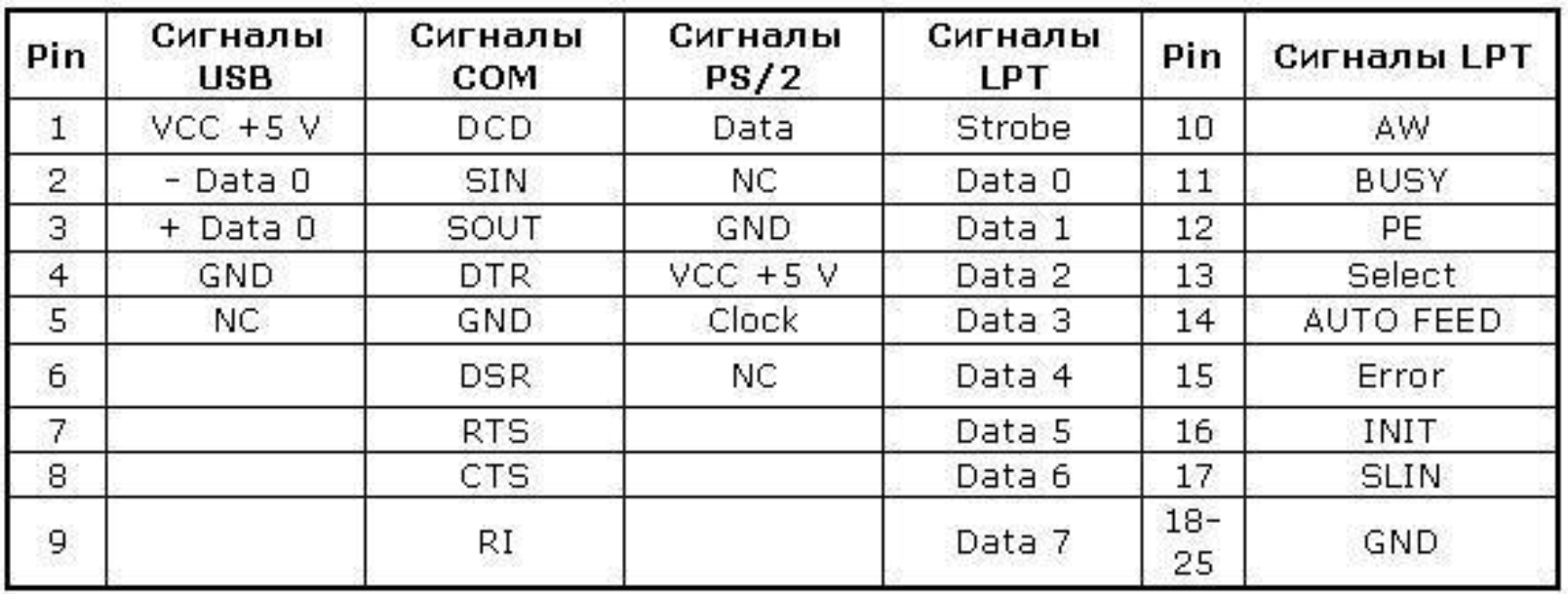
Инструменты повышения производительности отсутствуют

* 1. Средства диагностики оборудования

Неисправность в вычислительной технике можно определить по показателям, приведенным в таблицах:



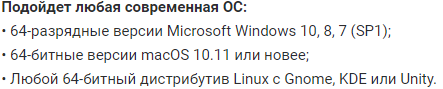


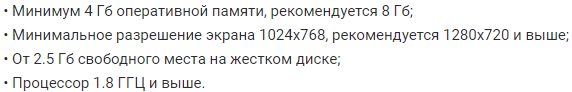


* 1. Измерение и анализ эксплуатационных характеристик качества программного обеспечения

Продукт доступен в двух версиях: PyCharm Community Edition — бесплатная версия, находится под лицензией Apache License, и PyCharm Professional Edition — расширенная версия продукта, обладающая дополнительной функциональностью, является проприетарным ПО.

Системные требования:





* 1. Оптимизация и модификация ПО

Приложение не требует дальнейшей оптимизации и/или модификации.

* 1. Разработка руководства оператора

1. Руководство пользователя должно быть организовано таким образом, чтобы любому человеку, не знакомому со средой программирования, программой и реализуемым ею методом, были понятны действия, которые от него требуется выполнить.
2. Примерный план написания руководства пользователя в соответствии с ГОСТ 19.505-79 «Руководство оператора» /2/:

* Действия при запуске программы (какой файл должен быть запущен, какие действия должны при этом выполняться).
* Структура программы (какие окна есть, как между ними переключаться, структура меню и назначение его элементов).
* Каким образом осуществляется ввод данных.
* Какая информация выводится на экран.
* Таким образом, нужно рассмотреть последовательность всех операций по работе с программой на конкретном примере, делая подробные комментарии и иллюстрируя текст графически.
* При необходимости можно указать: требуются ли от пользователя какие-либо настройки, если да, то какие; нужно ли использовать дополнительное программное обеспечение и т.п.

1. Этапы сопровождения и обслуживания ПО
   1. Разработка технического задания на сопровождение

|  |  |  |
| --- | --- | --- |
| Управленческий вопрос | Описание проблемы | Возможное решение |
| Согласование с организационными целями | возврат инвестиций от деятельности по сопровождению программного обеспечения | Программа должна быть универсальной и работать в долгой перспективе, также она должна обеспечить клиента всеми нужными функциями |
| Проблемы кадрового обеспечения | Нужен классифицированный персонал работников, который будет сопровождать ПО | Нужно нанять профессионалов, по сопровождению, которые понимаю свою работу и будут её выполнять верно. Для определения можно использовать тесты, такой вариант подходит если и команды есть достаточно средств для найма персонала. Либо есть второй вариант обучить новую команду сопровождения, минус этого варианта то-что это затратит больше времени |
| Процесс | Нужно найти ПО, которое поможет в сопровождение | Одна из наиболее детально проработанных и распространенных (на уровне стандарта de facto) процессных моделей, изначально созданных с ориентацией на программное обеспечение – CMMI. Уделяется особое внимание процессам сопровождения. |
| Организационные аспекты сопровождения | Нужно найти организацию для сопровождения | Организация занимающаяся сопровождением |
| Аутсорсинг | Нужно найти корпорацию, которая поможет в сопровождения ПО | Про анализировав все данные было решено передать часть работы компании-исполнителю. В их обязанности входят программные системы |

* 1. Обеспечение качества компьютерных систем в процессе эксплуатации

Обеспечение качества компьютерных систем в процессе эксплуатации является полноценным, если:

* Нужное ПО установлено
* Обоснован вариант конфигурации
* Обеспечен доступ различным категориям пользователей
* Обеспечена совместимость компонента с ранее установленными ПП
  1. Разработка сопровождающей документации

Пакет документов по сопровождению:

Документация по сопровождению ПС можно разбить на две группы:

(1) документация, определяющая строение программ и структур данных ПС и технологию их разработки;

(2) документацию, помогающую вносить изменения в ПС.

Документация первой группы содержит итоговые документы каждого технологического этапа разработки ПС. Она включает следующие документы:

* Внешнее описание ПС.
* Описание архитектуры ПС, включая внешнюю спецификацию каждой ее программы.
* Для каждой программы ПС - описание ее модульной структуры, включая внешнюю спецификацию каждого включенного в нее модуля.
* Для каждого модуля - его спецификация и описание его строения.
* Тексты модулей на выбранном языке программирования.
* Документы установления достоверности ПС, описывающие, как устанавливалась достоверность каждой программы ПС и как информация об установлении достоверности связывалась с требованиями к ПС.

Документация второй группы содержит:

* Руководство по сопровождению ПС, которое описывает известные проблемы вместе с ПС, описывает, какие части системы являются аппаратно- и программно-зависимыми, и как развитие ПС принято в расчет в его строении (конструкции).
* Договор на сопровождение с указанием процедур внесения изменений

4 Методы и средства защиты компьютерных систем

4.1 Тестирование на ошибки ввода/вывода данных

При работе в PyCharm может случиться:

* Ошибка недостатка памяти
* Ошибка недостатка ресурсов ПК

Способы решения:

* Очистите ваш ПК от ненужных файлов

4.2 Тестирование интерфейса

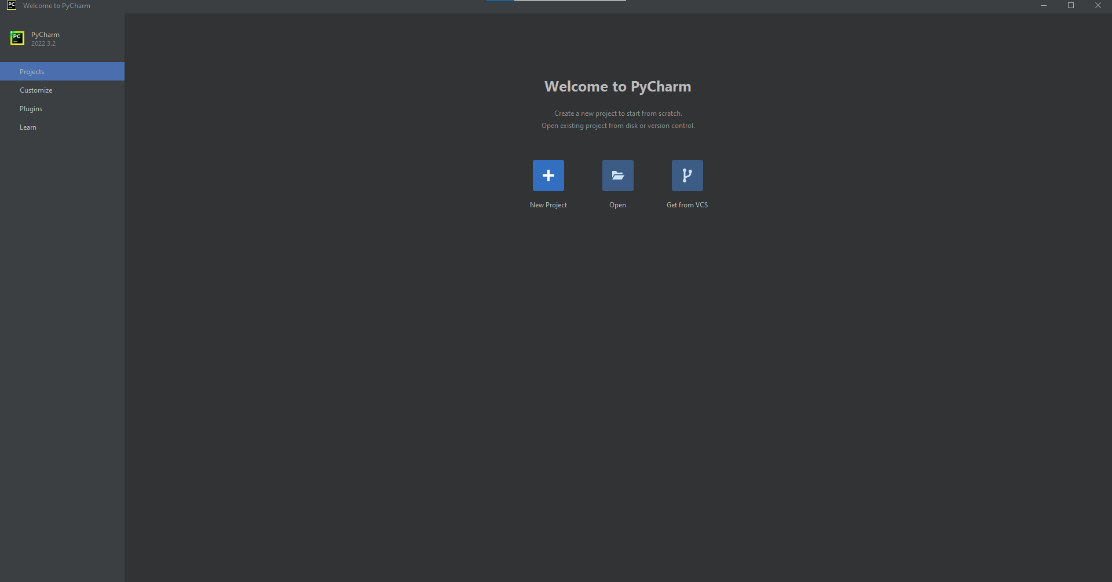


Рисунок 6 – интерфейс

Простой и понятный интерфейс. Похож на интерфейс IntelliJ idea, что достаточно удобно. Весь инструментарий расположен на интерфейсе сгруппировано.

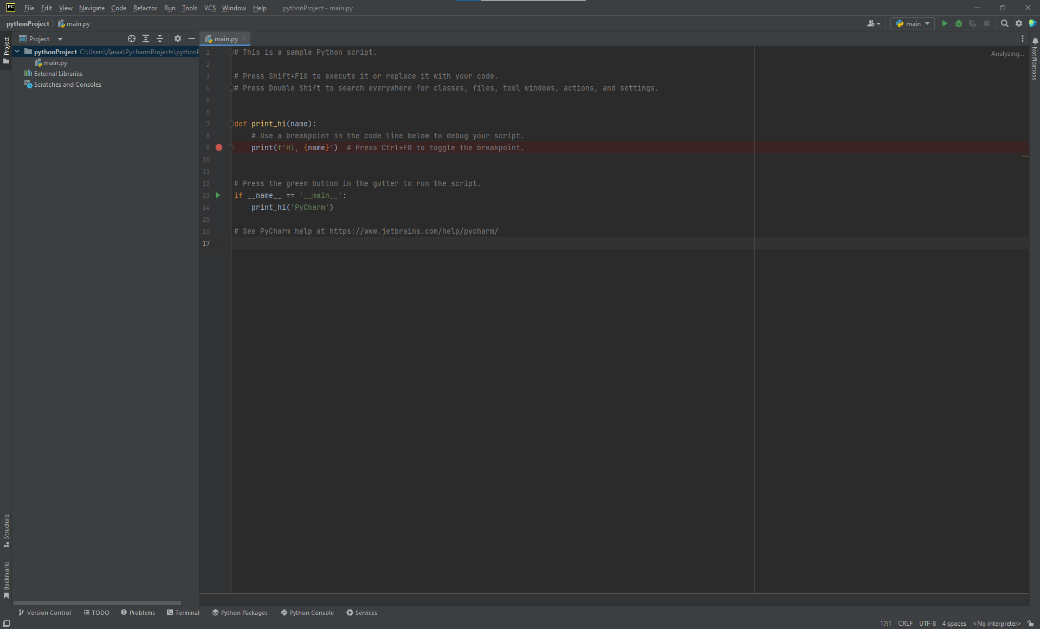


Рисунок 7 – запуск нового проекта

4.3 Тестирование контента

В PyCharm есть все инструменты, чтобы **писать, отлаживать и тестировать код**. Также IDE позволяет использовать в проектах другие языки программирования, синхронизировать код с системами контроля версий и развёртывать его.

4.4 Анализ рисков. Выявление первичных и вторичных ошибок

PyCharm не проверяет программы на вредоносность. При запуске «сломанной» программы существует возможность нанести вред системе пользователя.

4.5 Обнаружение вируса и устранение последствий его влияния

Рисунок 8 – логотип «Лаборатории Касперского»

Kaspersky

Все продукты «Лаборатории Касперского» для Windows отлично справляются с нейтрализацией вредоносного ПО и при этом не сильно влияют на производительность системы. А в программу начального уровня встроена даже защита от Ransomware.

4.6 Тестирование защиты программного обеспечения

У PyCharm нет защитных мер ни в каком виде.

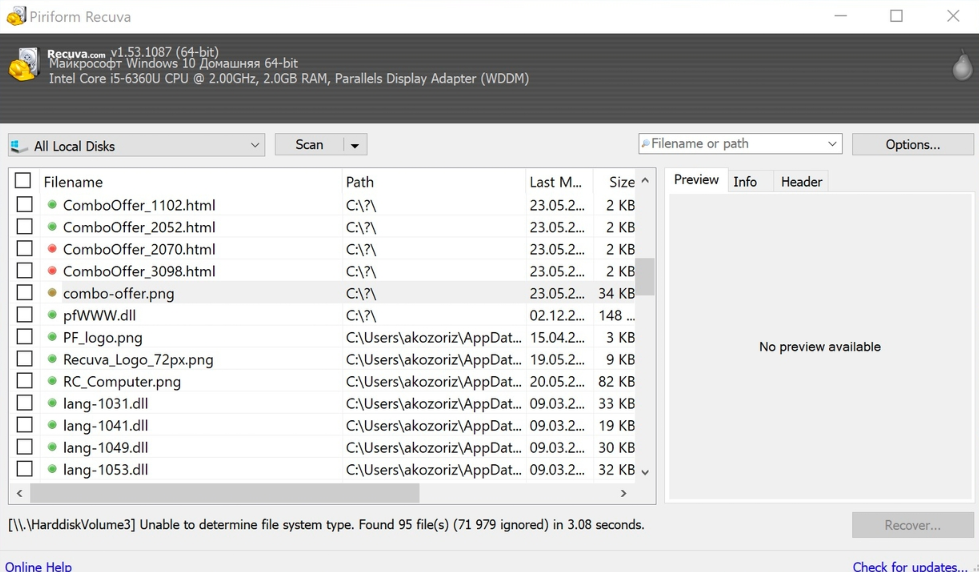
4.7 Работа с реестром. Работа с программой восстановления файлов и очистки дисков

Рисунок 9 – Recuva

Популярная программа с понятным интерфейсом и удобным мастером настройки восстанавливает данные с жёстких дисков, USB-флешек и карт памяти. В Recuva доступны простой режим для обычных пользователей и расширенный — для продвинутых. А благодаря портативной версии утилита легко запускается с флешки или диска без установки.

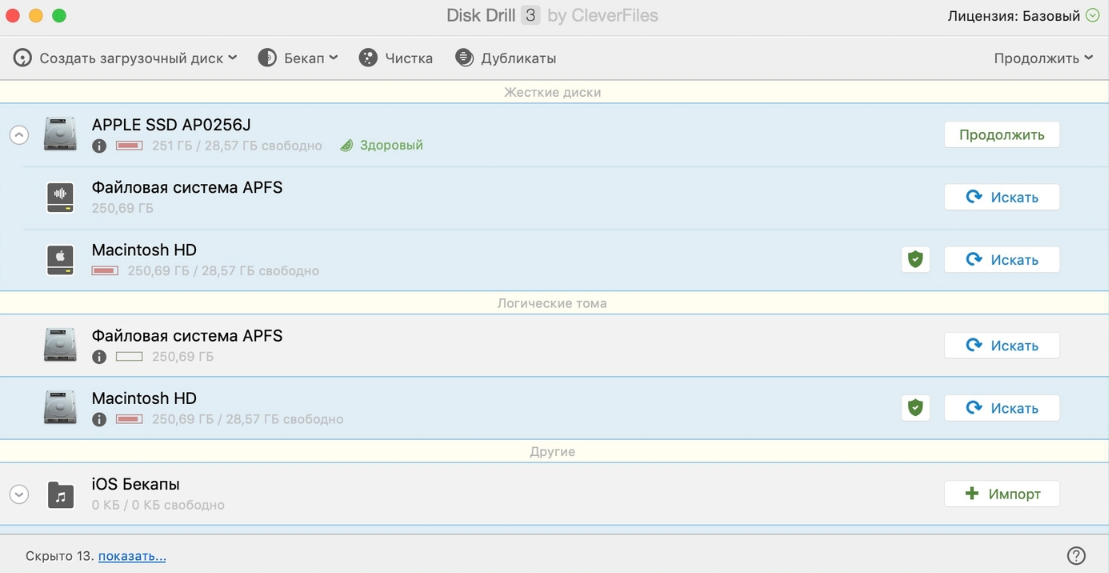


Рисунок 10 – Disk Drill

Функциональная утилита для Windows и macOS с лёгкостью воссоздаёт утраченные данные с самых разных носителей. Для экономии времени при сканировании фильтром можно задать определённые типы файлов, а после останется лишь выбрать нужные. У Disk Drill нет переносной версии, поэтому проблемный диск придётся подключать к другому компьютеру для восстановления.

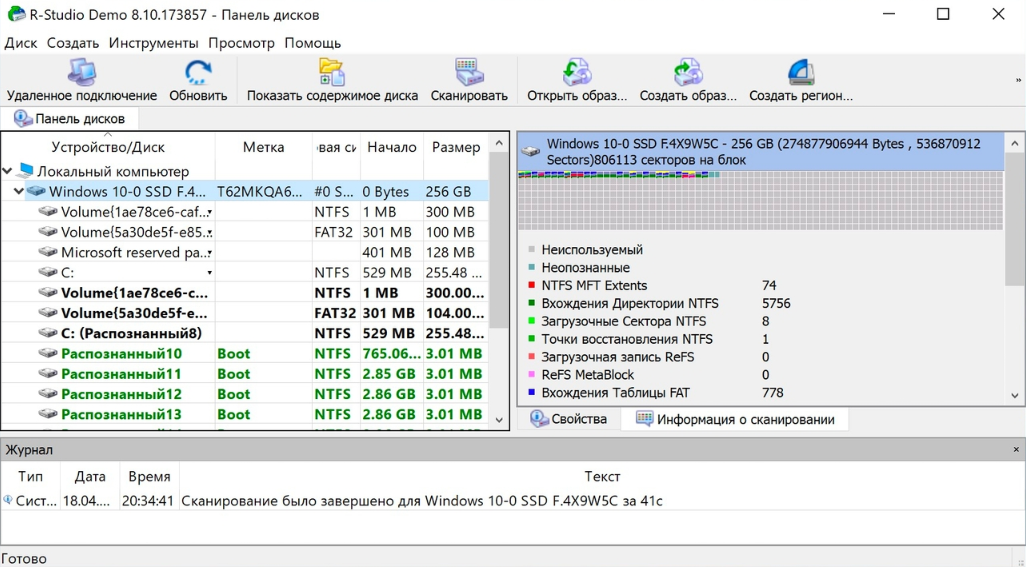


Рисунок 11 – R-Studio

Кроссплатформенная программа с богатыми возможностями. Благодаря продвинутым алгоритмам она способна восстановить данные с повреждённых и отформатированных разделов, внешних дисков и даже по сети. R-Studio умеет работать с загрузочного диска и поможет даже в том случае, когда ОС не запускается. Утилита позволяет создавать образы проблемных дисков и взаимодействовать с ними напрямую, повышая вероятность успешного восстановления данных.

5 Индивидуальное задание

5.1 Практические работы

Практическая работа №1 «Внедрение программного обеспечения»

Внедрение программного обеспечения — процесс настройки программного обеспечения под определенные условия использования, а также обучение пользователей работе с программным продуктом.

Внедрение программного продукта состоялось в том случае, если программный продукт выполняет поставленную задачу, а сотрудники компании полностью перешли на работу с новым продуктом.

Результатом проекта внедрения должен быть прозрачный, четко регламентированный, документированный и автоматизированный процесс разработки и сопровождения.

Основные этапы внедрения программного продукта:

1. Обследование

2. Разработка технического задания

3. Настройка системы

4. Тестирование системы

5. Опытная эксплуатация

6. Промышленная эксплуатация

Эффекты от внедрения можно разделить на две большие категории: эффект, получаемый на стратегическом уровне, т.е. на уровне бизнеса, и тактический эффект, который получат все участники процесса.

1. : Стратегический эффект

\* обеспечение прозрачности и измеримости достижения стратегических целей;

\* снижение вероятности проявления рисков в сфере информационных технологий;

\* повышение рентабельности IT-услуг за счет снижения сроков проведения проекта, снижения издержек поддержки и т.д.;

\* повышение инвестиционной привлекательности IT-проектов;

\* повышение доверия бизнеса к IT за счет качественно обработанных запросов IT-отделами от бизнеса.

2. : Тактический эффект

\* уменьшение сроков и снижение стоимости обработки новых запросов;

\* улучшение качества IT-услуг – это вовремя выполненные работы с заранее определенным качеством;

\* увеличение эффективности используемых ресурсов – хорошо организованный и формализованный процесс позволяет четко определить роли и ответственности каждого участника этого процесса;

\* более четкое и реалистичное планирование;

\* значительное уменьшение времени на принятие решения;

\* снижение влияния человеческого фактора.

Сопровождение (поддержка) программного обеспечения — процесс улучшения, оптимизации и устранения дефектов программного обеспечения (ПО) после передачи в эксплуатацию.

Сопровождение ПО — это одна из фаз жизненного цикла программного обеспечения, следующая за фазой передачи ПО в эксплуатацию.

Определение процесса сопровождения:

•SWEBOK: Сопровождение ПО – вся совокупность деятельности, необходимой для обеспечения эффективной поддержки программных систем.

•IEEE 1219 (Standard for Software Maintenance): Сопровождение ПО – определяется стандартом IEEE Standard for Software Maintenance (IEEE 1219) как модификация программного продукта после передачи в эксплуатацию для устранения сбоев, улучшения показателей производительности и/или других характеристик (атрибутов) продукта, или адаптации продукта для использования в модифицированном окружении.

•ГОСТ Р ИСО/МЭК 12207: Сопровождение – как процесс модификации программного продукта в части его кода и документации для решения возникающих проблем при эксплуатации или реализации потребностей в улучшениях тех или иных характеристик продукта.

Сопровождение поддерживает функционирование программного продукта на протяжении всего операционного жизненного цикла, то есть периода его эксплуатации.

В процессе сопровождения фиксируются и отслеживаются запросы на модификацию (также называемые “запросами на изменения” – change requests, в частности, в контексте конфигурационного управления), оценивается влияние предлагаемых изменений, производится модификация кода и других активов (артефактов) продукта, проводится необходимое тестирование и, наконец, выпускается обновленная версия продукта. Кроме того, проводится обучение пользователей и обеспечивается их ежедневная поддержка при работе с текущей версией продукта.

Практическая работа №2 «Техническая поддержка ПО»

Образец договора технической поддержки программного обеспечения.

1. Термины договора на сопровождение программ

1.1. Продукты – предоставленное по лицензионному договору № \_\_ от «\_\_»\_\_\_\_\_\_\_\_\_\_\_ 202\_ г. программное обеспечение.

1.2. Техническая поддержка - оказываемые услуги по настройке, обслуживанию, адаптации и модификации Продуктов или устранению имеющихся в них ошибок, а также предоставлению обновлений и дополнительных программных модулей, иные действия, предусмотренные в разделе 2 настоящего Договора.

1.3. Ошибка - дефект в коде Продукта, в результате которого данный Продукт не способен работать в соответствии функциональными возможностями, указанными в предоставленной на него технической документации, за исключением случаев:

(1) нарушения Заказчиком правил эксплуатации Продуктов в соответствии с требованиями предоставленной на них технической документации;

(2) использования Продуктов на оборудовании или совместно с программным обеспечением, которые не были рекомендованы Исполнителем.

2. Предмет договора техподдержки ПО

2.1. Исполнитель обязуется оказывать по заявкам Заказчика услуги по Технической поддержке, а Заказчик принимать и оплачивать оказываемые Исполнителем услуги.

2.2. Стороны согласовали возможность оказания услуг по Технической поддержке в следующем объеме и составе:

\* установка Продуктов на оборудование Заказчика;

\* настройка Продуктов на оборудовании Заказчика, включая их адаптацию;

\* модификация Продуктов по отдельному заказу;

\* предоставление выпускаемых обновлений Продуктов;

\* устранение ошибок в Продуктах;

\* консультирование по порядку использования Продуктов.

2.3. Услуги по Договору оказываются в соответствии с Соглашением об уровне услуг (SLA), являющимся неотъемлемой частью настоящего Договора.

Практическая работа №3 «Защита ПО»

Методы доказательства правильности программ могут быть применены для анализа безопасности ПО при существенных ограничениях на размеры и сложность создаваемых программ. Поэтому в частных случаях они могут оказаться более эффективными, чем другие известные методы анализа программ, которые исследуются в следующих разделах данной работы.

Методы, используемые для анализа и оценки безопасности ПО, разделяют на две категории: контрольно-испытательные и логико-аналитические. В основу данного разделения положены принципиальные различия в точке зрения на исследуемый объект (программу).

Контрольно-испытательные методы анализа рассматривают РПС через призму фиксации факта нарушения безопасного состояния системы, а логико-аналитические - через призму доказательства наличия отношения эквивалентности между моделью исследуемой программы и моделью РПС.

Контрольно-испытательные делятся на те, в которых контролируется процесс выполнения программы и те, в которых отслеживаются изменения в операционной среде, к которым приводит запуск программы.

При проведении анализа безопасности с помощью логико-аналитических методов строится модель программы и формально доказывается эквивалентность модели исследуемой программы и модели РПС. В простейшем случае в качестве модели

В целом полный процесс анализа ПО включает в себя три вида анализа:

\* лексический верификационный анализ;

\* синтаксический верификационный анализ;

\* семантический анализ программ

Каждый из видов анализа представляет собой законченное исследование программ согласно своей специализации.

Результаты исследования могут иметь как самостоятельное значение, так и коррелироваться с результатами полного процесса анализа.

Лексический верификационный анализ предполагает поиск распознавания и классификацию различных лексем объекта исследования (программа), представленного в исполняемых кодах. При этом лексемами являются сигнатуры. В данном случае осуществляется поиск сигнатур следующих классов:

\* сигнатуры вирусов;

\* сигнатуры элементов РПС;

\* сигнатуры (лексемы) "подозрительных функций";

\* сигнатуры штатных процедур использования системных ресурсов и внешних устройств.

Поиск лексем (сигнатур) реализуется с помощью специальных программ-сканеров.

Синтаксический верификационный анализ предполагает поиск, распознавание и классификацию синтаксических структур РПС, а также по-строение структурно-алгоритмической модели самой программы.

Решение задач поиска и распознавания синтаксических структур РПС имеет самостоятельное значение для верификационного анализа программ, поскольку позволяет осуществлять поиск элементов РПС, не имеющих сигнатуры. Структурно-алгоритмическая модель программы необходима для реализации следующего вида анализа - семантического.

Семантический анализ предполагает исследование программы изучения смысла составляющих ее функций (процедур) в аспекте операционной среды компьютерной системы. В отличие от предыдущих видов анализа, основанных на статическом исследовании, семантический анализ нацелен на изучение динамики программы - ее взаимодействия с окружающей средой. Процесс исследования осуществляется в виртуальной операционной среде с полным контролем действий программы и отслеживанием алгоритма ее работы по структурно-алгоритмической модели.

Семантический анализ является наиболее эффективным видом анализа, но и самым трудоемким. По этой причине методика сочетает в себе три перечисленных выше анализа. Выработанные критерии позволяют разумно сочетать различные виды анализа, существенно сокращая время исследования, не снижая его качества.

Практическая работа №4.

|  |  |
| --- | --- |
| Тип комплектующих | Модель комплектующих |
| Процессор | AMD Ryzen 5 5600x |
| Оперативная память | 32 гб DIMM DDR4 |
| Видеокарта | AMD Radeon RX 6600 |
| Звуковая плата | Интегрированный контроллер |
| Сетевая карта | Интегрированный контроллер |
| Жёсткий диск | 1Tb |
| Дисковод для гибких дисков | нет |
| Оптический привод | нет |
| Клавиатура | SVEN KB-G9600 |
| Мышь | A4 Tech Bloody V8 |
| Монитор | ASUS VP228DE |
| Корпус | ZET GAMING Rare |
| Блок питания | MIRAGE GOLD 650W |
| Источник бесперебойного питания | нет |

5.2 Проектирование проведения электронных выставок и покупательских конференций

* Выбор и анализ темы.
* Определение цели назначения.
* Поиск документов по теме.
* Выявление и отбор документов.
* Разработка структуры выставки и выбор формы.
* Составление библиографических характеристик на представленные документы
* Сканирование иллюстраций, набор текста, оцифровка документов.
* Выбор формата выставки.
* Размещение текста на слайдах и его форматирование, т.е изменение шрифта, его размера, эффекта и цвета.
* Размещение картинок, иллюстраций, аудио, видео и других материалов.
* Создание анимации
* Создание навигации
* Предварительный просмотр
* Исправление недостатков и ошибок
* Размещение на сайте
* Информирование зрителей и реклама выставки
* Анализ эффективности выставки

5.3 Характеристики программ с точки зрения влияния на их защищенность и результаты работы

Основными характеристиками программявляются:

* алгоритмическая сложность (логика алгоритмов обработки информации);
* состав и глубина проработки реализованных функций обработки;
* полнота и системность функций обработки;
* объем файлов программ;
* требования к операционной системе и техническим средствам обработки со стороны программного средства;
* объем дисковой памяти;
* размер оперативной памяти для запуска программ;
* тип процессора;
* версия операционной системы;
* стоимость и др.

Защита программного обеспечения преследует цели:

* ограничение несанкционированного доступа к программам или их преднамеренное разрушение и хищение;
* исключение несанкционированного копирования (тиражирования) программ.

Программный продукт и базы данных должны быть защищены по нескольким направлениям от воздействия:

1) человека– хищение машинных носителей и документации ПО; нарушение работоспособности программного продукта и др.;

2) аппаратуры– подключение к компьютеру аппаратных средств для считывания программ и данных или их физического разрушения;

3) специализированных программ– приведение программного продукта или базы данных в неработоспособное состояние, несанкционированное копирование программ, баз данных и т.д.

Одним из эффективных способов защиты программных продуктов и баз данных является ограничение доступа. Контроль доступа данных строится путем:

* парольной защиты программ при их запуске;
* использования ключевой дискеты для запуска программ;
* ограничения прав доступа пользователей к программам или данным, функциям обработки;
* использования криптографических методов защиты информации и т.п.

Заключение

В заключение хочется сказать, что потраченного времени жаль. Но также стоит отметь и то, что в ходе проделанной работы были выявлены плюсы работы с PyCharm, а также был приобретён опыт работы с данной средой разработки.

Источники

<https://pythonchik.ru/osnovy/faq-po-rabote-s-pycharm>

<https://skillbox.ru/media/code/pycharm-kak-eye-ustanovit-i-ispolzovat/>

<https://ru.wikipedia.org/wiki/PyCharm>

<https://lifehacker.ru/vosstanovlenie-dannyx-s-zhyostkogo-diska/>