Задания для домашней контрольной работы по дисциплине «Основы программной инженерии».

Задание 1.

Сравнение систем управления версиями ПО.

Система управления версиями (VCS), также известная как система управления исходным кодом, – это программное обеспечение для отслеживания изменений в файловой системе и управления ими. VCS также предлагает средства для совместной работы, которые позволяют обмениваться этими изменениями файловой системы и привязывать их к другим пользователям VCS. При работе на уровне файловой системы VCS отслеживает добавление, удаление и изменение файлов и каталогов. Репозиторий – это понятие, связанное с VCS, место, в котором хранится история отслеживания файловой системы VCS. Применительно к отдельным файлам исходного кода VCS отслеживает добавления, удаления и изменения строк текста в таких файлах. Популярными системами управления версиями являются Git, Mercurial, SVN и Preforce.

Пользу VCS невозможно переоценить. Этот инструмент вносит массу преимуществ в процесс совместной работы команды разработчиков. В любом проекте по разработке ПО, в котором над файлами исходного кода работает не один разработчик, обязательно нужно использовать VCS. Кроме того, проектам, в которых код сопровождает один специалист, VCS тоже поможет. По сути, в современном проекте по разработке ПО не существует веской причины отказываться от использования VCS.

Средства VCS используют две основные модели работы: централизованную и распределенную.

При сравнении достоинств и недостатков каждой модели ключевым вопросом становится возможность создания удаленной резервной копии. У централизованной VCS точка отказа одна — удаленный экземпляр централизованной VCS. При утрате этого экземпляра может снизиться продуктивность и могут быть потеряны данные. Его нужно будет заменить другой копией исходного кода. Если экземпляр становится временно недоступным, разработчике не смогут отправлять изменения, выполнять их слияние или откатывать код. Архитектура, использующая распределенную модель, лишена этих недостатков, поскольку полная копия исходного кода хранится в каждом экземпляре VCS. Если возникает какая-либо проблема из вышеописанных с распределенной VCS, новый экземпляр VCS может стать основным для разработки, и таким образом снижения производительности можно избежать.

Интеграция VCS в проект по разработке ПО открывает доступ к множеству преимуществ для организации и управления. Обычная система VCS сама по себе предлагает ранее описанные технические средства разрешения командных конфликтов и совместной работы. Размещенная VCS отличается от обычной усовершенствованными возможностями. Эта усовершенствованная VCS обладает целым арсеналом средств и дает наглядное представление о процессе разработки ПО, который было принято считать творческим поиском «вслепую».

Задание 2.

Приведения типов данных.

Преобразование числовых типов

В Python существует два числовых типа данных: целые числа и числа с плавающей точкой. Для преобразования целых чисел в числа с плавающей точкой и наоборот Python предоставляет специальные встроенные методы.

Преобразование целых чисел в числа с плавающей точкой

Метод float() преобразовывает целые числа в числа с плавающей точкой. Число указывается в круглых скобках:

float(57)

Это преобразует число 57 в 57.0.

Также можно использовать переменные. Объявите переменную f = 57, а затем выведите число с плавающей точкой:

f = 57

print(float(f))

57.0

Преобразование чисел с плавающей точкой в целые числа

Встроенная функция int() предназначена для преобразования чисел с плавающей точкой в целые числа.

Функция int() работает так же, как и float(). Чтобы преобразовать число, добавьте его в круглые скобки:

int(390.8)

Число 390.8 преобразуется в 390.

Эта функция также может работать с переменными. Объявите переменные:

b = 125.0

c = 390.8

Затем преобразуйте и отобразите их:

print(int(b))

print(int(c))

125

390

Чтобы получить целое число, функция int() отбрасывает знаки после запятой, не округляя их (потому 390.8 не преобразовывается в 391).

Преобразование чисел с помощью деления

При делении Python 3 может преобразовать целое число в число с плавающей точкой (в Python 2 такой функции нет). К примеру, разделив 5 на 2, вы получите 2.5.

a = 5 / 2

2.5

Python не преобразовывает тип данных во время деления; следовательно, деля целое число на целое число, в результате вы получили бы целое число, 2.

Преобразование строк

Строка – это последовательность из одного или больше символов (цифр, букв и других символов). Строки – очень распространённый тип данных в программировании. Существует много способов преобразования строк.

Преобразование чисел в строки

Чтобы конвертировать число в строку, используйте метод str(). Поместите число или переменную в круглые скобки.

Попробуйте преобразовать целое число, например:

str(12)

Запустив метод str(12) в интерактивной оболочке Python (с помощью команды python), вы получите вывод:

'12'

Преобразование строк в числа

Строки можно преобразовать в числа с помощью методов int() и float()

Если в строке нет десятичных знаков, лучше преобразовать её в целое число. Для этого используется int().

Преобразование в кортежи и списки

Чтобы преобразовать данные в кортеж или список, используйте методы tuple() и list() соответственно. В Python:

Список – это изменяемая упорядоченная последовательность элементов, взятая в квадратные скобки ([ ]).

Кортеж – неизменяемая упорядоченная последовательность элементов, взятая в круглые скобки.

Преобразование списка в кортеж

Преобразовывая список в кортеж, вы можете оптимизировать программу. Для преобразования в кортеж используется метод tuple().

print(tuple(['pull request', 'open source', 'repository', 'branch']))

('pull request', 'open source', 'repository', 'branch')

Выведенные на экран данные являются кортежем, а не списком, поскольку они взяты в круглые скобки.

Преобразование в списки

Вы можете преобразовать кортеж в список, чтобы сделать его изменяемым.

Обратите внимание: при этом в методах list() и print() используется две пары круглых скобок. Одни принадлежать собственно методу, а другие – кортежу.

print(list(('blue coral', 'staghorn coral', 'pillar coral')))

['blue coral', 'staghorn coral', 'pillar coral']