



Программа профессиональной пробы

Разработчик программного обеспечения: создание
простого приложения с графическим интерфейсом
на языке Python

Профессия:

Инженер-программист

УГСН/УГПС:

27.00.00 Управление в технических системах

Профессиональная среда:

Деловая среда

Область профессиональной деятельности:

6 Связь, информационные и коммуникационные технологии

Вид профессиональной пробы:

Базовый

Продолжительность:

90 минут

Классы:

10, 11

Доступность для участников с ОВЗ и/или инвалидностью:

Общие заболевания

Место проведения:

г Калининград, Советский пр-кт, д 1

Автор:

Романов Михаил Андреевич

2025 г.

Введение (10 минут)

Краткое описание профессии

Разработчик программного обеспечения (ПО) — это специалист, который создает, тестирует и поддерживает программные продукты: от веб-сайтов и мобильных приложений до сложных операционных систем и игр. Его работа включает анализ потребностей пользователя, проектирование архитектуры программы, написание кода на одном из языков программирования (таких как Python, Java, C#), отладку и устранение ошибок. Это творческая и логическая профессия, требующая постоянного обучения и умения работать в команде. Разработчик превращает идеи в функциональные и удобные диджитал-решения.

Место и перспективы профессионального направления в современной экономике региона, страны, мира

ИТ-сектор является одним из наиболее динамично растущих в мировой экономике, определяя развитие всех остальных отраслей — от логистики и сельского хозяйства до медицины и образования. В России спрос на квалифицированных разработчиков стабильно высок. В Калининградской области, являющейся анклавом с особой экономической зоной, ИТ-сфера имеет особый потенциал. Здесь работают как локальные студии (например, по разработке игр), так и удаленные команды крупных российских и международных компаний. Развитие региона как цифрового хаба поддерживается на государственном уровне, что создает благоприятные условия для старта и построения карьеры в ИТ без необходимости переезда в другие города.

Необходимые навыки и знания для овладения профессией (с указанием рекомендованного класса)

Базовые знания алгоритмов и структур данных (что такое переменная, цикл, условие, список).

Знание основ одного из языков программирования (желательно Python, как одного из самых простых для старта).

Логическое и алгоритмическое мышление, умение разбивать сложную задачу на простые шаги.

Базовые знания английского языка (для чтения технической документации).

Внимательность, усидчивость и готовность к постоянному самообучению.

Креативность для поиска нестандартных решений.

1-2 интересных факта о профессиональном направлении

Язык программирования Python был назван не в честь змеи, а в честь британского комедийного сериала «Летающий цирк Монти Пайтона». Создатель языка Гвидо ван Россум был большим поклонником шоу.

Первым в мире программистом была женщина — Ада Лавлейс. В середине XIX века она составила первую в мире программу для аналитической машины Чарльза Бэббиджа, хотя сама машина так и не была построена.

Связь профессиональной пробы с реальной деятельностью

Участники пробы выполняют реальную задачу frontend-разработчика — создание графического интерфейса (GUI) для программы. Это ключевой этап в разработке любого пользовательского приложения, будь то калькулятор, мессенджер или сложный аналитический инструмент. Школьники узнают, как код взаимодействует с визуальными элементами (кнопками, полями ввода), что является основой для создания современных удобных программ.

Цели и задачи (ожидаемые результаты) профессиональной пробы

Сформировать у школьников начальное представление о профессии разработчика ПО через практическое создание простого графического приложения на Python.

Обучающие:

- Познакомить с базовыми принципами работы графического интерфейса пользователя (GUI).
- Освоить основные элементы библиотеки Tkinter (окно, кнопка, метка, поле ввода).
- Закрепить навыки написания чистого и структурированного кода на Python.
- Научить связывать интерфейсные элементы с логикой программы (обработка событий).

Развивающие:

- Развивать алгоритмическое и логическое мышление.
- Развивать навыки поиска и исправления ошибок (дебаггинг).
- Стимулировать креативность при проектировании внешнего вида приложения.

Воспитательные:

- Воспитывать внимательность, аккуратность и perseverance (настойчивость) в достижении цели.
- Показать возможности для профессионального роста в ИТ-сфере в Калининградской области.

Формы и методы

Форма проведения: Практико-ориентированное занятие в компьютерном классе с элементами мини-лекции.

Методы:

1. Объяснительно-иллюстративный: Краткая презентация о профессии и основах библиотеки Tkinter.
2. Репродуктивный: Выполнение пошаговых инструкций педагога по созданию базовых элементов интерфейса.
3. Частично-поисковый (проблемный): Самостоятельное решение типовых ошибок и задач по модификации кода.
4. Практический метод: Индивидуальная работа за компьютером по созданию собственного приложения.
5. Рефлексия: Коллективное обсуждение результатов, трудностей и успехов.

Инструктаж по технике безопасности

1. Общие требования:

- К работе допускаются учащиеся, прошедшие данный инструктаж.
- Запрещается трогать провода питания и подключения оборудования.
- Запрещается работать на неисправном оборудовании (с запахом гари, дыма, треснувшим монитором).

2. Перед началом работы:

- Убедитесь в исправности оборудования. Обо всех неполадках немедленно сообщите педагогу.
- Разместите на столе тетрадь, учебное пособие так, чтобы они не мешали работе за компьютером.
- Сядьте прямо напротив монитора, соблюдайте расстояние от глаз до экрана не менее 50-70 см.

3. Во время работы:

- Соблюдайте правильную осанку: спина прямая, плечи -расслаблены, ноги стоят на полу.
- Избегайте резких движений, не кладите рядом с клавиатурой жидкости и посторонние предметы.
- Следите за равномерным освещением рабочего места, избегайте бликов на экране.
- При появлении рези в глазах, головной боли, неисправности оборудования немедленно прекратите работу и сообщите педагогу.
- Не нажимайте одновременно много клавиш на клавиатуре.

4. В аварийных ситуациях:

- При возникновении пожара, запаха гари или задымления немедленно отключите оборудование и покиньте кабинет, сообщив педагогу. Действуйте по плану эвакуации.
- При получении травмы или резком ухудшении самочувствия обратитесь к педагогу.

5. По окончании работы:

- Закройте все активные программы.
- Аккуратно приведите рабочее место в порядок.
- С разрешения педагога покиньте кабинет.

Постановка задачи (5 мин)

Постановка профессионального задания

Участникам профессиональной пробы предстоит, используя язык Python и библиотеку Tkinter, самостоятельно разработать и запрограммировать простое приложение с графическим интерфейсом пользователя (GUI) по предложенному шаблону.

Итоговый результат, продукт

Конечным продуктом работы каждого школьника является работоспособная программа с графическим интерфейсом, например, «Калькулятор» с «Конвертером величин».

Программа должна включать:

1. Главное окно приложения с заголовком.
2. Визуальные элементы (виджеты): поля для ввода данных, кнопки, текстовые метки с инструкциями.
3. Реализованную логику: программа должна корректно обрабатывать нажатия кнопок, выполнять заложенные математические операции (сложение, вычитание, конвертацию по заданному коэффициенту) и выводить понятный результат пользователю.
4. Интуитивно понятный интерфейс, элементы которого расположены упорядоченно и логично.

Участник не только создаст исполняемый файл (.py), но и получит представление о полном цикле создания элементарного ПО: от проектирования интерфейса до написания кода и тестирования готового продукта.

Выполнение задания (55 минут)

Подробная инструкция по выполнению задания

Этап 1: Подготовка и знакомство с интерфейсом (15 минут)

1. Создать новый файл с расширением .py (например, my_app.py).
2. В первой строке файла импортировать библиотеку для создания интерфейса:
`from tkinter import *`

Этап 2: Создание основного окна (10 минут)

3. Создать главное окно программы: `window = Tk()`
4. Задать заголовок окна с помощью метода `window.title("Мой Калькулятор")`.
5. Задать размер окна: `window.geometry("300x200")` (ширина x высота).

Этап 3: Добавление элементов управления (30 минут)

6. Создать два поля для ввода чисел. Используйте виджет `Entry()` и разместите его на форме с помощью метода `.grid(row=0, column=1)`.
7. Добавить текстовые метки (`Label`) рядом с полями ввода, например, "Число 1:" и

"Число 2:".

8. Создать кнопку (Button) с текстом "Сложить". Используй команду `command=add_numbers` (функцию `add_numbers` ты создашь на следующем этапе).

9. Создать метку (Label) для вывода результата, например, `result_label = Label(window, text="Результат:")`.

Этап 4: Написание логики программы (30 минут)

10. Написать функцию `add_numbers()`, которая будет срабатывать при нажатии на кнопку.

11. Внутри функции:

- * Получить числа из полей ввода с помощью метода `.get()`.
- * Преобразовать полученные текстовые значения в числа с помощью `float()`.
- * Выполнить сложение этих чисел.
- * Обновить текст в `result_label` на результат вычисления, используя `result_label.config(text="Результат: " + str(sum))`.

Этап 5: Запуск и тестирование (15 минут)

12. В конце файла добавить строку `window.mainloop()`, которая запускает цикл обработки событий окна.

13. Запустить программу. Проверить её работу: ввести числа в оба поля, нажать кнопку и убедиться, что правильный результат появился на экране.

14. Исправить ошибки, если они возникли.

Оценка результатов выполнения задания (5 минут)

Критерии успешного выполнения задания

1. Техническая реализация (работоспособность):

Программа запускается, не требует исправлений для запуска. Все заявленные элементы интерфейса (поля ввода, кнопка, метка результата) присутствуют и отображаются корректно.

2. Функциональность и логика:

Программа корректно выполняет основную операцию (сложение чисел). Результат вычисления выводится четко и ясно. Обработка события нажатия на кнопку реализована верно.

3. Качество кода и оформление:

Код чистый, читаемый, с осмысленными именами переменных. Отсутствуют "мертвые" или закомментированные без причины участки кода. Логика разделена на функции.

Рекомендации для наставника по контролю результата, процедуре оценки

Непрерывный мониторинг: Основная часть оценки происходит в процессе работы. Наблюдайте за тем, как участник подходит к решению задачи, как ищет ошибки, насколько он самостоятелен. Это дает более точную картину, чем только

финальный результат.

Финальная демонстрация: Попросите каждого участника кратко (1-2 минуты) продемонстрировать свою работающую программу. Это развивает навык презентации результата — ключевой навык для разработчика.

Проверка кода: Бегло просмотрите код на предмет соответствия критериям качества. Не требуйте идеального стиля, но обратите внимание на грубые нарушения.

Обсуждение (15 минут)

Вопросы

1. В чем заключаются основные задачи представителя профессии?
2. Какие особенности условий труда характеризуют данную профессию?
3. Какие качества необходимы представителю данной профессии?
4. Понравилась ли Вам деятельность, которой вы занимались на профессиональной пробе? Захотелось ли Вам узнать о профессии больше?
5. Возможен ли выбор Вами данного направления профессиональной деятельности?
6. С какими основными трудностями вы столкнулись при выполнении задания?
7. Как вы думаете, почему Python часто рекомендуют как первый язык для изучения программирования?

Инфраструктурный лист для проведения профессиональной пробы

Наименование	Рекомендуемые технические характеристики с необходимыми примечаниями	Кол-во	Кол-во пользователей
Ноутбук или персональный компьютер с монитором, мышью и клавиатурой	Стандартный компьютер или ноутбук с установленными интерпретатором Python, средой разработки (IDE) или текстовым редактором	1	На человека