**Методические указания**

**Проект «Сертификация сотрудников»**

**🚀СЮЖЕТНАЯ ЛИНИЯ:**

В прошлый раз к специалистам ProTeam обратилось сразу четыре заказчика: АО «БЭСК», ПАО «МТС», Segezha group, агрохолдинг «Степь». Каждому из них необходимо разработать приложение для тестирования знаний по технике безопасности сотрудников. АО «БЭСК» необходимо протестировать электриков, ПАО «МТС» — офисных сотрудников, Segezha group — лесорубов, а в агрохолдингe «Степь» — механизаторов сельского хозяйства. Каждый заказчик предоставил список вопросов тестирования и материалы для оформления теста и обозначил дедлайн — через 2 недели.

В прошлый раз разработчики выбрали себе заказчиков, спланировали работу над проектами, запрограммировали интерфейсы приложений и сформировали на основе данных заказчика списки вопросов теста. Сегодня им предстоит реализовать функционал тестирования и презентовать результат работы.

**⚠️ РЕЗЮМЕ:**

**Цель** — применить на практике навыки создания приложений и презентации результатов работы для завершения проекта.

Этот урок является последним в курсе Python Start, поэтому крайне важно создать ситуацию успеха у каждого ученика. К концу занятия у всех должны получиться работающие приложения, затем результаты работы должны быть презентованы группами учеников заказчикам.

**💾 ССЫЛКИ И РЕКВИЗИТ:**

* [презентация](https://docs.google.com/presentation/d/1wBuZXRQGGe3edmpl1fxnxOOPmY9vxWWXEm_wIXaAI0E/edit?usp=sharing), [документация](https://backoffice.algoritmika.org/task-preview/33099?track=1&position=1), [презентация](https://docs.google.com/presentation/d/14Llolx74EwZwl3GGJjDjr8ZiFcpm3e1foGpp_XKhs18/edit?usp=sharing) для защиты проектов.

# 

# 🎯 ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ УРОКА

| *После урока ученик:* | | *Результат достигнут, когда ученики:* |
| --- | --- | --- |
| * использует наработки прошлых проектов для реализации функционала; * программирует переключение между панелями интерфейса (между панелью вопроса и панелью результатов); * программирует подсчёт и отображение статистики прохождения теста; * самостоятельно тестирует и оценивает готовое приложение; * защищает результат работы перед заказчиком с помощью презентации. | | * участвовали в обсуждениях и задавали уточняющие вопросы; * запрограммировали функции, связанные с функционалом тестирования и подсчётом его результата; * завершили работу над приложением; * защитили работы перед заказчиками. |

**РЕКОМЕНДУЕМАЯ СТРУКТУРА УРОКА**

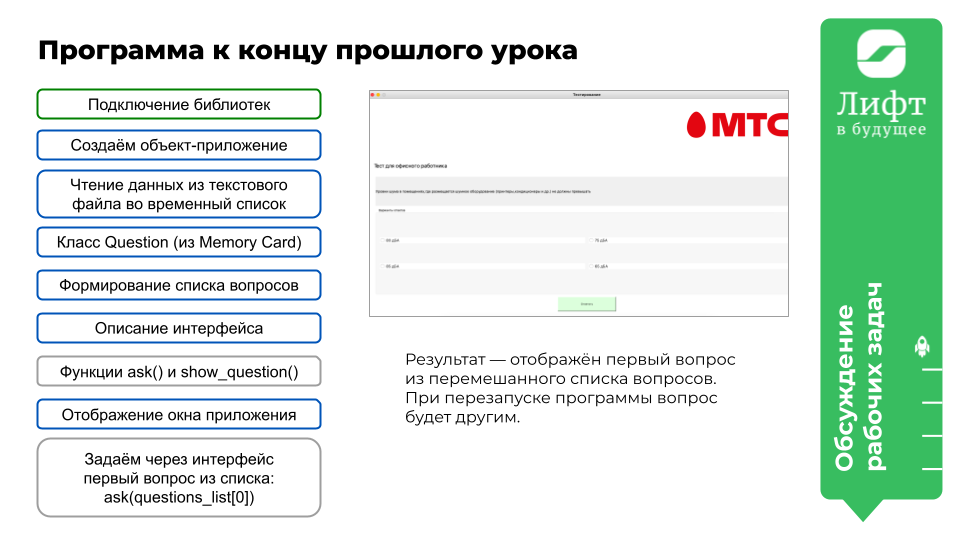
| **Время** | **Этап** | Задачи этапа |
| --- | --- | --- |
| 5 мин | Сюжетная линия. Обсуждение:  «Сертификация сотрудников» | * Напомнить сюжетную задачу: разработать универсальное приложение для тестирования сотрудников для четырёх заказчиков. * С помощью mind map и чек-листа рабочих задач вспомнить, какие задачи уже выполнены, а какие нужно выполнить сегодня. * Анонсировать презентацию работ во второй половине занятия. |
| 15 мин | «Мозговой штурм»:  «Функционал тестирования» | * Описать функционал, который осталось запрограммировать. * Обратиться к полезным наработкам из приложения Memory Card. * Обсудить создание функций, связанных с переключением между вопросами и отображением результатов теста. |
| 25 мин | Платформа:  «VSC. Сертификация сотрудников» | * Поставить задачу: завершить приложение по сертификации знаний. * Самостоятельно запустить и протестировать приложение. |
| 5 мин | Перерыв | * Провести разминку, либо организовать смену деятельности детей. |
| 20 мин | «Мозговой штурм»:  «Презентация работы»  и  Подготовка к презентации | * Анонсировать презентацию проектов перед всеми заказчиками. * Предложить ученикам презентовать приложения в мини-группах. * Изучить заготовку презентации, распределить слайды выступлений и назначить ответственных за представление приложений. * Заполнить слайды и постараться успеть один раз отрепетировать выступление |
| 15 мин | Презентация  проектов | * Пригласить зрителей, объявить начало презентации. * Объяснить, какая задача стояла перед разработчиками. Рассказать о решении работать в мини-группах. * Передать слово ученикам и предложить поделиться результатами работы. |
| 5 мин | Завершение урока. Рефлексия | * Объявить об окончании контракта с компанией ProTeam. Дать обратную связь ученикам об их работе. * Предложить ребятам поделиться впечатлениями от изучения Python. |

# Сюжетная линия. Обсуждение: «Сертификация сотрудников»

##### *(5 мин)*

Откройте презентацию. Разработчикам компьютеры пока не требуются.

| *«Здравствуйте, коллеги! Продолжаем работу над заказами на разработку приложений от крупных российских компаний.*  *Для АО «БЭСК» необходимо разработать приложение для оценки знаний по технике безопасности у электриков, для ПАО «МТС» — у офисных сотрудников, для Segezha group — у лесорубов, а для агрохолдинга «Степь» — у механизаторов сельского хозяйства. Каждый заказчик предоставил нам материалы для работы. Срок работы истекает уже сегодня!*  *В прошлый раз мы запрограммировали бОльшую часть проекта, пользуясь наработками одного из прошлых приложений Memory Card. Сегодня мы должны дописать функционал, связанный с тестированием, и презентовать результат заказчикам».* |
| --- |



С помощью слайдов напомните разработчикам содержание заказа и спланированную структуру проекта. Назовите уже реализованные части проекта и покажите их на mind map. Обозначьте задачи на текущий рабочий день и отметьте их в чек-листе.

Сформулируйте цель и переходите к «мозговому штурму».

# «Мозговой штурм»: «Функционал тестирования»

##### *(15 мин)*

С помощью слайда рассмотрите ожидаемый функционал приложения. На настоящий момент у разработчиков должно быть готово приложение со стилем заказчика, которое считывает список вопросов из файла, при запуске отображает случайный вопрос и даёт выбрать ответ. Кнопка «Ответить»/«Закрыть» пока не активна, переключения между вопросами и с панели вопроса на панель результата нет.

Чтобы разобраться, как именно запрограммировать недостающий функционал, предложите разработчикам посмотреть на приложение глазами пользователя. Что отобразится, если пользователь запустит программу сейчас? Какое действие он выполнит следующим? Как программа должна «откликнуться» на это действие?

Воспользуйтесь тем, что подобные функции с небольшими отличиями разработчики уже программировали для проекта Memory Card. По итогам обсуждения придите к необходимости запрограммировать ряд функций:

* **Функция-обработчик клика по кнопке**. В зависимости от состояния программы на кнопке может быть одна из надписей: «Ответить» или «Закрыть». Если пользователь кликнул по кнопке «Ответить», то должен отобразиться следующий вопрос (случай, когда этот вопрос последний, лучше рассмотреть в другой функции), а если по «Закрыть», то тестирование завершено и окно приложения должно закрыться.
* **Функция проверки правильности ответа на вопрос.** Вместе с кликом по кнопке «Ответить» может происходить и проверка ответа на вопрос. В приложении для сертификации ответы не должны показываться сразу, как в Memory Card. Вместо этого после прохождения всего теста пользователю должна выводиться статистика правильных ответов. Для этого необходимо завести два счётчика: для подсчёта количества заданных вопросов и количества правильных ответов.

# 

По аналогии рассмотрите ещё две функции, связанные с **переключением на следующий вопрос** и **отображением результата тестирования**.

Подведите итоги обсуждения, поставьте задачу на первую половину урока и переходите к работе в VS Code.

# 

# Платформа: «VSC. Проект “Сертификация сотрудников”»

##### *(25 мин)*

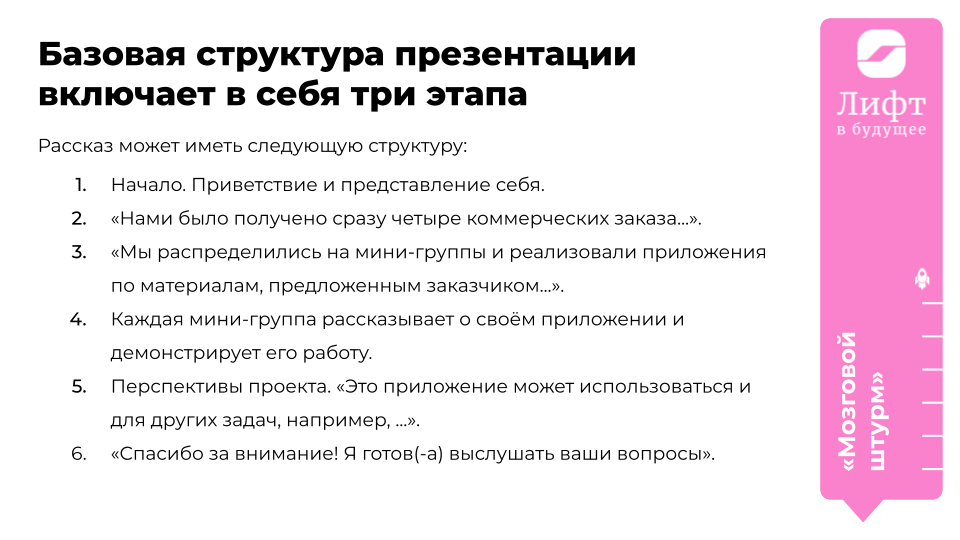
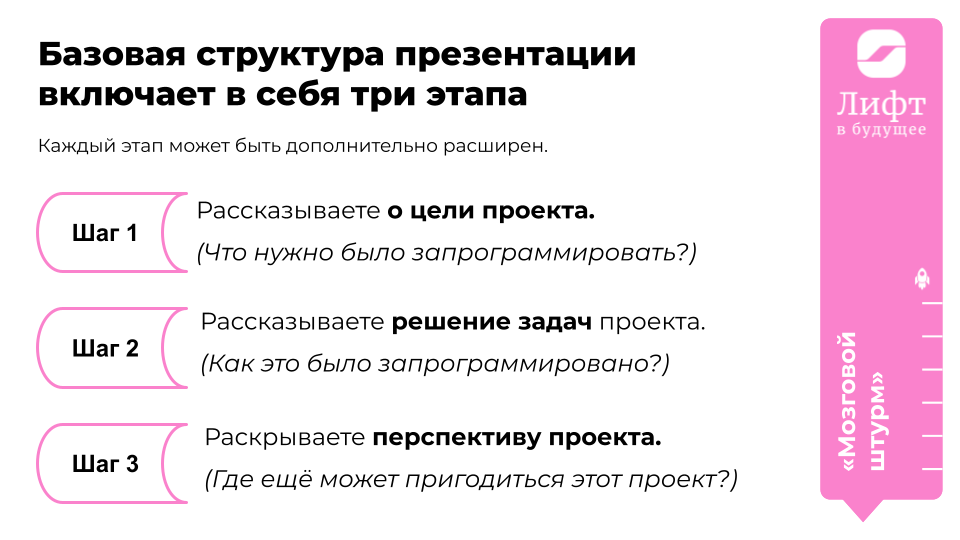
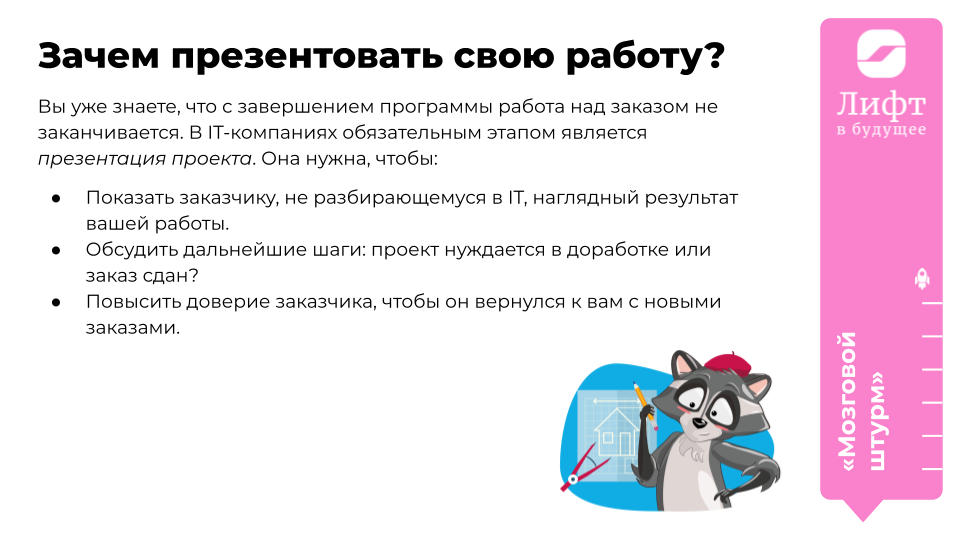
Организуйте работу в VS Code. Ученикам необходимо дописать часть функционала, связанную с тестированием и отображением результата. К концу этапа у всех учеников должно быть полностью готово приложение для выбранного ими заказчика (включая стилизацию).

# «Мозговой штурм»: «Презентация работы» и создание презентации

##### *(20 мин)*

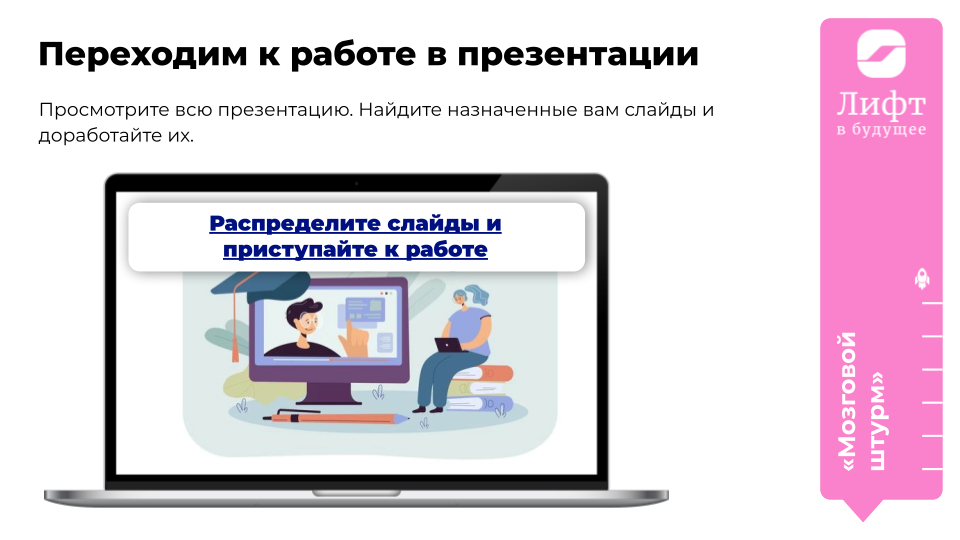
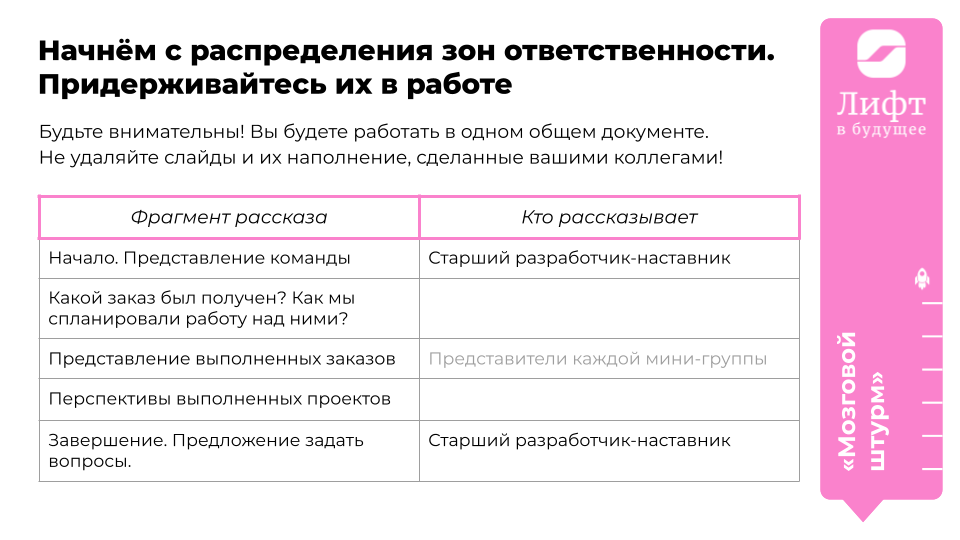
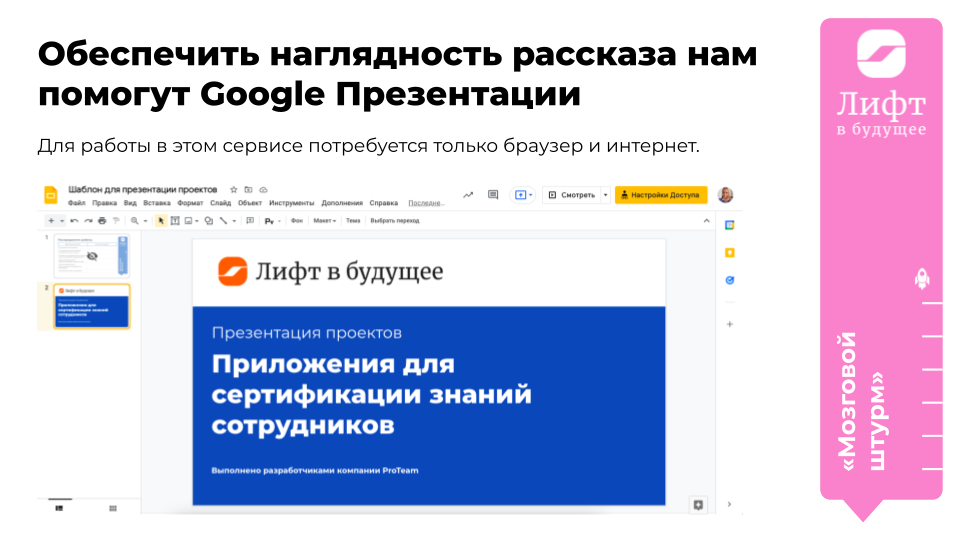
##### Напомните разработчикам, что с завершением программы работа над проектом не заканчивается: заказчику нужно презентовать разработанное программное обеспечение. При этом важно обеспечить наглядность и говорить с аудиторией «на одном языке», тогда довольный заказчик не раз вернётся с новыми предложениями о сотрудничестве!

##### Продемонстрируйте разработчикам базовую структуру презентации проекта и заготовку предложений, которые можно использовать при выступлении. Отметьте, что из-за общего назначения приложений (всероссийская акция) доклад о проделанной работе тоже будет общим.



Опорные точки доклада и иллюстрации можно разместить на слайдах с помощью Google Презентаций. Познакомьте разработчиков с основами работы с презентациями. Обратите внимание, что работа будет идти в общем документе, поэтому важно быть внимательными и не редактировать слайды коллег. Для этого будут распределены зоны ответственности, которых нужно придерживаться.

**Методический комментарий.** Для экономии времени на уроке воспользуйтесь готовым шаблоном презентации (необходимо сделать копию). Распределите между учениками блоки выступления, связанные с демонстрацией приложений, а начало и конец доклада возьмите на себя.



Откройте копию шаблона презентаций, вышлите ссылку разработчикам через марсограм или чат онлайн-платформы. Перейдите в первый скрытый слайд и распределите слайды. Затем сразу переходите к работе над презентацией.

К концу этого этапа у вас должна быть готова презентация для выступления. Каждый ученик должен представлять, какую часть информации он представит на презентации.

# Презентация проектов

##### *(15 мин)*

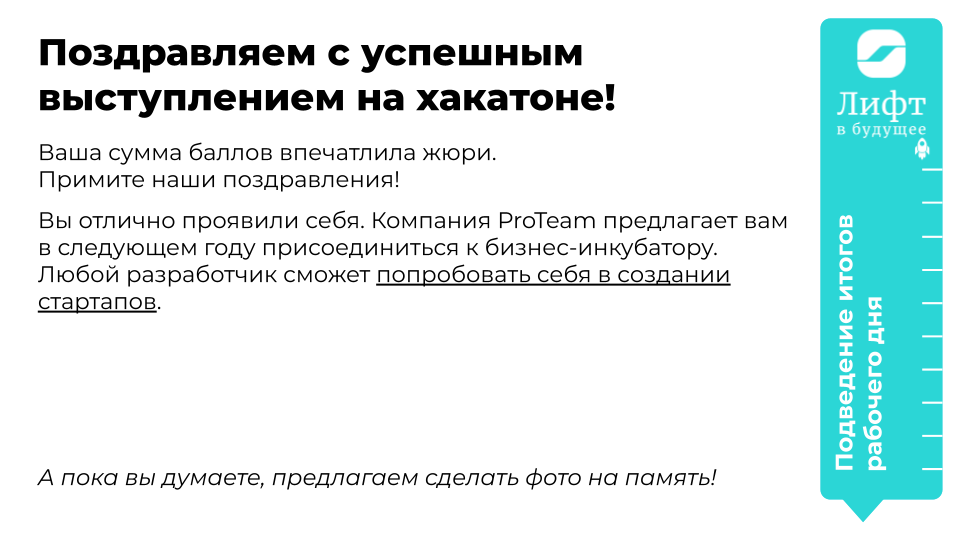
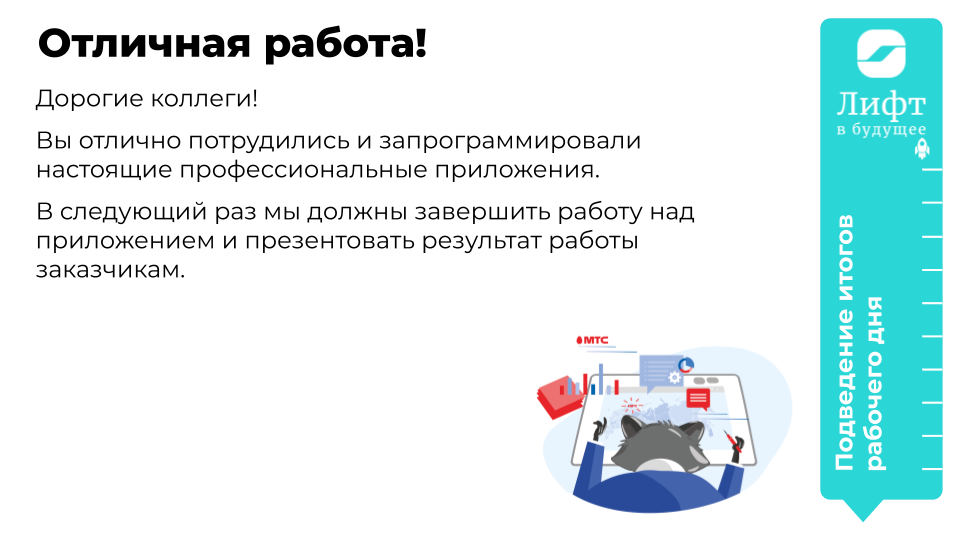
Пригласите в кабинет гостей последнего урока курса и объявите начало презентации. После небольшого вступления с рассказом о поставленной перед разработчиками задаче передайте слово ребятам.

Заключительное слово презентации также за вами. Поблагодарите гостей за внимание и сообщите о готовности выслушать их вопросы и пожелания.

# Завершение урока

##### *(5 мин)*

Объявите о завершении контракта с компанией ProTeam и поздравьте разработчиков с успешной сдачей всех проектов. Выразите уверенность, что работа с Python на этом не заканчивается, и у каждого сотрудника есть возможность погрузиться в определённую сферу, например, в анализ данных или создание IT-стартапов.



Расспросите ребят об их впечатлениях и дайте каждому позитивную обратную связь. Предложите сделать фото на память.

**Ответы на задания**

**1. Текст программы с готовым функционалом**

from PyQt5.QtCore import Qt

from PyQt5.QtGui import \*

from PyQt5.QtWidgets import \*

from random import randint, shuffle

#импортируем модуль с цветом и логотипом

#from mts import \*

from steppe import \*

#from bashkirenergo import \*

#from segezha\_group import \*

# открытие файла и чтение файла

all\_item\_list = []

with open(file\_name, encoding='UTF-8') as file:

test\_name = file.readline() # первая строчка файла - название теста

for line in file:

all\_item\_list.append(line.strip()) # построчно считываем файл

class Question():

''' содержит вопрос, правильный ответ и три неправильных'''

def \_\_init\_\_(self, question, right\_answer, wrong1, wrong2, wrong3):

self.question = question

self.right\_answer = right\_answer

self.wrong1 = wrong1

self.wrong2 = wrong2

self.wrong3 = wrong3

# заполняем список вопросов объектами класса Question

questions\_list = []

i = 0

while i < len(all\_item\_list):

questions\_list.append(Question(all\_item\_list[i], all\_item\_list[i+1], all\_item\_list[i+2], all\_item\_list[i+3], all\_item\_list[i+4] ))

i = i + 5 # шаг 5 элементов, т.к 1 вопрос + 4 варианта ответа

shuffle(questions\_list) #перемешиваем список вопросов

app = QApplication([])

#с помощью setStyleSheet задаём стиль виджетам

btn\_OK = QPushButton('Ответить') # кнопка ответа

btn\_OK.setStyleSheet(ans\_button\_style)

btn\_OK.setFixedSize(200, 50) # устанавливаем фиксированный размер виджетам

lb\_Question = QLabel("Вопрос") # текст вопроса

lb\_Question.setWordWrap(True) #перенос строки вопроса

lb\_Test = QLabel(test\_name) # название теста

lb\_Question.setFixedSize(1450, 100)

lb\_Question.setStyleSheet(q\_text\_style)

lb\_Test.setStyleSheet(header\_text\_style)

RadioGroupBox = QGroupBox("Варианты ответов") # группа на экране для переключателей с ответами

RadioGroupBox.setFixedSize(1450, 300)

RadioGroupBox.setStyleSheet(radio\_group\_style)

Res\_list = QListWidget() # виджет для вывода списка вопросов и правильных ответов в конце теста

Res\_list.hide()

Res\_list.setStyleSheet(lw\_res\_list\_style)

rbtn\_1 = QRadioButton('Вариант 1')

rbtn\_2 = QRadioButton('Вариант 2')

rbtn\_3 = QRadioButton('Вариант 3')

rbtn\_4 = QRadioButton('Вариант 4')

pixmap = QPixmap(logo) #создаём объект-картинку

logo = QLabel() # создаём контейнер под объект-картинку

logo.resize(pixmap.width(), pixmap.height()) # меняем размер контейнера под картинку

logo.setPixmap(pixmap) # добавляем объект-картинку

rbtn\_1.setStyleSheet(radiobutton\_style)

rbtn\_2.setStyleSheet(radiobutton\_style)

rbtn\_3.setStyleSheet(radiobutton\_style)

rbtn\_4.setStyleSheet(radiobutton\_style)

RadioGroup = QButtonGroup() # для группировки переключателей, чтобы управлять их поведением

RadioGroup.addButton(rbtn\_1)

RadioGroup.addButton(rbtn\_2)

RadioGroup.addButton(rbtn\_3)

RadioGroup.addButton(rbtn\_4)

layout\_ans1 = QHBoxLayout()

layout\_ans2 = QVBoxLayout() # вертикальные будут внутри горизонтального

layout\_ans3 = QVBoxLayout()

layout\_ans2.addWidget(rbtn\_1) # два ответа в первый столбец

layout\_ans2.addWidget(rbtn\_2)

layout\_ans3.addWidget(rbtn\_3) # два ответа во второй столбец

layout\_ans3.addWidget(rbtn\_4)

layout\_ans1.addLayout(layout\_ans2)

layout\_ans1.addLayout(layout\_ans3) # разместили столбцы в одной строке

RadioGroupBox.setLayout(layout\_ans1) # готова "панель" с вариантами ответов

AnsGroupBox = QGroupBox("Результат теста")

AnsGroupBox.setStyleSheet(result\_group\_style)

lb\_Finish = QLabel("Тест окончен")

lb\_Finish.setStyleSheet(lb\_Finish\_style)

lb\_Result = QLabel("") # здесь будет размещен результат теста(количество правильных ответов)

lb\_Result.setStyleSheet(lb\_Result\_style)

layout\_res = QVBoxLayout()

layout\_res.addWidget(lb\_Finish, alignment=(Qt.AlignLeft | Qt.AlignTop))

layout\_res.addWidget(lb\_Result, alignment=(Qt.AlignCenter | Qt.AlignTop))

AnsGroupBox.setLayout(layout\_res)

layout\_line0 = QHBoxLayout()

layout\_line1 = QHBoxLayout() # вопрос и картинка

layout\_line2 = QHBoxLayout() # варианты ответов или результат теста

layout\_line3 = QHBoxLayout() # кнопка "Ответить"

layout\_line0.addWidget(lb\_Test,alignment=(Qt.AlignLeft| Qt.AlignBottom))

layout\_line0.addWidget(Res\_list,alignment=Qt.AlignLeft)

layout\_line0.addWidget(logo, alignment=Qt.AlignRight)

layout\_line1.addWidget(lb\_Question,alignment=Qt.AlignLeft)

layout\_card = QVBoxLayout()

layout\_line2.addWidget(RadioGroupBox)

layout\_line2.addWidget(AnsGroupBox)

AnsGroupBox.hide() # скроем панель с ответом, сначала должна быть видна панель вопросов

layout\_line3.addStretch(1)

layout\_line3.addWidget(btn\_OK)

layout\_line3.addStretch(1)

layout\_card.addLayout(layout\_line0,stretch=2)

layout\_card.addLayout(layout\_line1,stretch=2)

layout\_card.addLayout(layout\_line2, stretch=4)

layout\_card.addStretch(1)

layout\_card.addLayout(layout\_line3, stretch=1)

layout\_card.addStretch(1)

layout\_card.setSpacing(2) # пробелы между содержимым

def show\_question():

''' показать панель вопросов '''

btn\_OK.setText('Ответить')

RadioGroup.setExclusive(False) # сняли ограничения, чтобы можно было сбросить выбор радиокнопки

rbtn\_1.setChecked(False)

rbtn\_2.setChecked(False)

rbtn\_3.setChecked(False)

rbtn\_4.setChecked(False)

RadioGroup.setExclusive(True) # вернули ограничения, теперь только одна радиокнопка может быть выбрана

answers = [rbtn\_1, rbtn\_2, rbtn\_3, rbtn\_4]

def ask(q: Question):

''' функция записывает значения вопроса и ответов в соответствующие виджеты,

при этом варианты ответов распределяются случайным образом'''

shuffle(answers) # перемешали список из кнопок, теперь на первом месте списка какая-то непредсказуемая кнопка

answers[0].setText(q.right\_answer) # первый элемент списка заполним правильным ответом, остальные - неверными

answers[1].setText(q.wrong1)

answers[2].setText(q.wrong2)

answers[3].setText(q.wrong3)

lb\_Question.setText(q.question) # вопрос

show\_question() # показываем панель вопросов

'''ПЕРЕКЛЮЧЕНИЕ МЕЖДУ ВОПРОСАМИ И ОТОБРАЖЕНИЕ РЕЗУЛЬТАТОВ ТЕСТА'''

def check\_answer():

''' проверка ответа на правильность '''

if answers[0].isChecked():

window.score += 1

next\_question()

else:

if answers[1].isChecked() or answers[2].isChecked() or answers[3].isChecked():

next\_question()

def next\_question():

''' задает вопрос из списка вопросов '''

window.total += 1

if window.total <= len(questions\_list):

q = questions\_list[window.total - 1] # взяли вопрос

ask(q) # спросили

else:

show\_result()

def show\_result():

''' функция показывает результаты теста'''

btn\_OK.setText("Закрыть")

btn\_OK.setFixedSize(300, 50)

btn\_OK.setStyleSheet(res\_button\_style)

RadioGroupBox.hide()

lb\_Question.hide()

AnsGroupBox.show()

lb\_Result.setText('Всего вопросов: '+str(window.total-1)+ '\n\nПравильных ответов: '+str(window.score))

def click\_OK():

''' определяет, что делать в зависимости от надписи на кнопке'''

if btn\_OK.text() == 'Ответить':

check\_answer() # проверка ответа

elif btn\_OK.text() == 'Закрыть':

exit()

window = QWidget()

window.setStyleSheet(background)

window.setLayout(layout\_card)

window.setWindowTitle("Тестирование")

btn\_OK.clicked.connect(click\_OK) # по нажатии на кнопку выбираем, что конкретно происходит

window.score = 0

window.total = 0

next\_question()

window.setFixedSize(win\_width, win\_height)

window.show()

app.exec()