УДК 004.432

Обзор языка программирования Python

Осмонова Бермет Майрамбековна
Нарынский государственный университет С.Нааматова
Преподаватель кафедры «Информационные технологии»
Приамурский государственный университет им. Шолом-Алейхема
магистрант

Аннотация
В настоящее время одним из самых популярных языков программирования является Python. Python – это язык общего назначения. Он может применяться в различных сферах IT: от создания простого приложения до реализации систем на основе машинного обучения. Решение подобных задач, как правило, сопровождается разработкой графического интерфейса пользователя (GUI), для чего может быть использовано несколько библиотек [5]. В работе описана основная информация о языке программирования Python. Кратко представлена история создания языка, сферы ее применения и актуальность на сегодняшний день. Рассматривается характерные особенности библиотек Python и их краткое описание. Также пример использование PyQt5, который является одним из наиболее часто используемых модулей для создания GUI приложений , PyQt5 Designer, благодаря которому можно создавать сложные GUI приложения в Python. Приводятся код написания разработки маленького приложения на PyQt5 и основная панель работы PyQt5 Designer.
Ключевые слова: Python, библиотеки для программирования на Python, PyQt5, Qt Designer, SublimeText, PyCharm.

An overview of the Python programming language

Osmonova Bermet Mairambekovna
Naryn State University after named S. Naamatov
Lector of the department "Information Technology"
Sholom-Aleichem Priamursky State University
master student

Abstract
Describes the basic information about the Python programming language. The history of the creation of the language is briefly presented, using and relevance for today. The characteristic features of Python libraries and their brief description are considered. Also an example of using PyQt5, which is one of the most commonly used modules for building GUI applications, PyQt5 Designer, which can create complex GUI applications in Python. The codes for developing a small application in PyQt5 and the main panel of the PyQt5 Designer are given.

Keywords: Python, Python programming libraries, PyQt5, Qt Designer, SublimeText, PyCharm.

Научный руководитель:
Баженов Руслан Иванович
Приамурский государственный университет имени Шолом-Алейхема
К.п.н., доцент, зав. кафедрой информационных систем, математики и правовой информатики

Введение

Актуальность исследования. В современном обществе во многих учебных учереждениях (средняя школа, средне-специальное, высшее, послевузовское) используют, развивают обучение с использованием языка программирование Python, можно найти онлайн обучающие платформы, краткосрочные курсы которые дают информацию о том, что этот язык очень популярен и кроссплатформен. Проанализируем возможности и практическое использование языка, его историю, основные библотеки и модули.

Обзор исследований.

Происхождение, преимущества и перспективы развития исследовали О.М.Голошубова, В.Ю. Наумов и описали ее историю создание, сравнили с другими языками программирование, выделили преимущества языка. Рассмотрели философию, модули расширение, универсальность языка и скорость разработки программы[1].Также М.А.Копытова в работе написала о языке, его истории и особенностях[4]. Начальные сведения о программировании на языке Python С. К. Буйначев, Н. Ю. Боклаг обширно дали для специальностей технического направления. [10]. В своей книге Beginning Python. From Novice to Professional автор Magnus Lie Hetland дал практическое руководство по ключевым функциям, преимущества методов исключения и абстракции, парадигмы программирования, отличный и легкий в обучении книга[11].

Цель исследования является выявить и определить особенности языка Python и библиотек. Рассмотрение и создание приложение на PyQt5 Designer.

Методы исследования.

Язык Python, анализ и разработка приложение на PyQt5 Designer.

Изначально все корни Python растут из процедурного структурного и высокоуровневого языка ABC, что задумывался как платформа-язык обучения программированию, по подобию языков Pascal и Бейсик. В разработке же языка ABC и принимал участие родоначальник и создатель языка Python – Гвидо Ван Россум (Guido Van Rossum), которому хотелось придумать не только простой и понятный язык, как ABC, который никак не использовался, кроме как в обучении начинающих программистов, но, и чтобы он был полезен для множества различных людей[1].

Создание Python началась в декабре 1989 года в центре математики и информатики в Нидерландах, хотя это и не совсем верно, так по словам

самого Гвидо Ван Россума «… в декабре 1989 я был в поисках проекта-хобби по программированию, что занял бы меня в течение недели Рождественских каникул. Мой офис был закрыт, но у меня дома был компьютер, и ничего более». Так что, исходя из его слов, можно считать, что начало Python было заложено в уютной, домашней и рождественской обстановке. Хотя сам Гвидо Ван Россум и задумал Python ещё в начале 1980-х, но кто знает, что было бы с Python и был бы он таким, каким мы видим его сейчас, если бы не его «моральный» предшественник ABC…? Практически Python создавался, как попытка исправить ошибки, допущенные при проектировании ABC[1].

Основными источниками вдохновения при создании Python было множество языков программирования, но я постараюсь выделить из них несколько основных, часто упоминаемых:

- ABC – использование отступов, вместо скобочек, для группировки операторов и относительная простота и лёгкость понимания кода.
- Modula-3 – первый язык, использовавший конструкцию исключений try и except, и подаривший её не только Python, но и такому языку как Java. Так же Python унаследовал от Modula-3 систему модулей и пакетов, хотя сама Modula-3 была не первым языком, в котором применялись модули.
- C, C++ - некоторые синтаксические конструкции. По словам самого Гвидо Ван Россума – он постарался выбрать самые непротиворечивые и логичные конструкции из C, чтобы не вызвать неприязнь C-программистов к Python.
- Lisp – отдельные черты функционального программирования.

Ещё один интересный момент состоит в том, что в начале своей жизни разработка Python была профинансирована правительством США, и собственно цель этого проекта была в том, чтобы создать язык для людей, чтобы этим людям в среднем было просто понять его и в дальнейшем начать на нём писать. Кроме того, стоит упомянуть, что Гвидо Ван Россуму присвоен шуточный статус «Великодушного пожизненного диктатора для языка Python», и это значит, что Гвидо до конца своей жизни руководит тем, как Python развивается и, в общем, никакие решения касательно судьбы языка без Гвидо принять не могут [1].

Рассмотрим известные направления применения Python:

- ✓ веб-разработка - для создании Python-фреймворков, например django, flask используются в серверной части приложений;
- ✓ data science: машинное обучение, анализ данных и визуализация - реализует некоторый алгоритм, который позволяет автоматически обнаруживать знакомый шаблон среди входных данных. Популярные алгоритмы машинного обучения это, нейронные сети; глубокое обучение; метод опорных векторов; random forest;
- ✓ автоматизация процессов - это написание небольших скриптов для автоматизации различных рабочих операций и процессов[2].

Что это, и где можем использовать?

- Отличительная черта Питона – это простой синтаксис, который делает его высокоуровневым языком и идеальным для новичков, а также тем, кто уже профессионально занимается программированием.
- Python – здесь имеется огромное количество модулей и библиотек под любые задачи. С помощью этой программы разрабатывают стартапы и используют платформы Instagram, YouTube.
- Также применятся в разработке мобильных приложений под Android, iOS, игр, в web-разработке, в машинном обучении, анализ данных, в образовании и т.д.

Одним из сильных сторон языка Питон является стандартные библиотеки, которые имеют средства для работы со многими сетевыми форматами и протоколами веб. Библиотека – это набор модулей, функций облегчающих специфические операции с использованием программирования.

Рассмотрим основные библиотеки, о которых должны знать разработчики.

1. Keras – нейросетевая библиотека, на сервере использует Theano и TensorFlow. Также предлагает некоторые из лучших функций для компиляции моделей, обработки наборов данных и визуализации графиков, исследования глубокого обучения.
2. NumPy - техническая вычислительная библиотека. Имеет открытый, бесплатный исходный код, облегчает вычисление сложных математических задач, простой в использовании и легко в кодировании.
3. Pillow – библиотека для обработки изображений. Отличительная черта библиотеки – улучшение качества изображения, размытия, яркость, контур, прозрачность и поддержка многих форматов изображений.
4. PYGLET – для разработки игр. Библиотека многоплатформенного кадрирования и мультимедии для Python, используется для визуально насыщенных приложений. Поддерживает все форматы видео, звука и изображений.
5. Requests - библиотека HTTP, направлена на то, чтобы сделать запросы HTTP проще и удобнее. Автоматически декодирует контент, поддерживает прокси-серверы и выполняет аутентификацию.
6. TensorFlow - библиотека машинного обучения, предназначенная для решения ряда задач, связанных с потоком данных и дифференцируемым программированием. Поддерживает обучение нескольких нейронных сетей и нескольких графических процессоров для создания эффективных моделей в крупных системах. Используется для нейронных сетей[3].

PyQt5 и Qt Designer.

PyQt5 - это набор Python библиотек для создания графического интерфейса на базе платформы Qt5 от компании Digia. PyQt – это библиотека Python с открытым исходным кодом для виджет-инструментария Qt, который также функционирует как кроссплатформенная среда разработки приложений. Qt – это популярная среда C++ для написания приложений с помощью графического интерфейса для всех основных настольных, мобильных и встраиваемых платформ [5, с. 362][5]. Графический интерфейс пользователя, или Graphical User Interface (GUI), представляет собой систему средств для взаимодействия пользователя с компьютером, основанную на возможности доступа к системным объектам и функциям в виде графических компонентов экрана, например, окон, значков, меню, кнопок, списков. Манипулировать GUI можно при помощи клавиатуры или мыши[6].

PyQt является одним из наиболее часто используемых модулей для создания GUI-приложений в Python. Одной из особенностей приведенной библиотеки является возможность использования Qt Designer (дизайнер GUI), благодаря которому можно создавать сложные интерфейсы, перетаскивая нужные виджеты для создания нужной формы. Прежде, чем создавать приложение на PyQt, требуется установить библиотеку PyQt и дизайнер Qt Designer в систему [7][5].

Для использования PyQt5 и Qt Designer можно зайти на официальный сайт и установить командой pip install PyQt5.

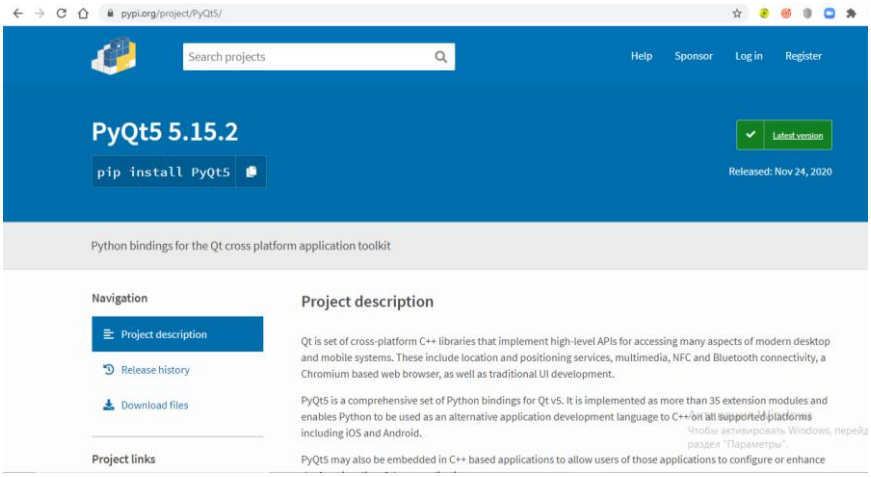

Рис.1. Официальный сайт pypi.org

Для установки используем командную строку, запуск от имени администратора, и вводим команду. После чего начинается инсталляция

pip install pyqt5, pip install PyQt5Designer

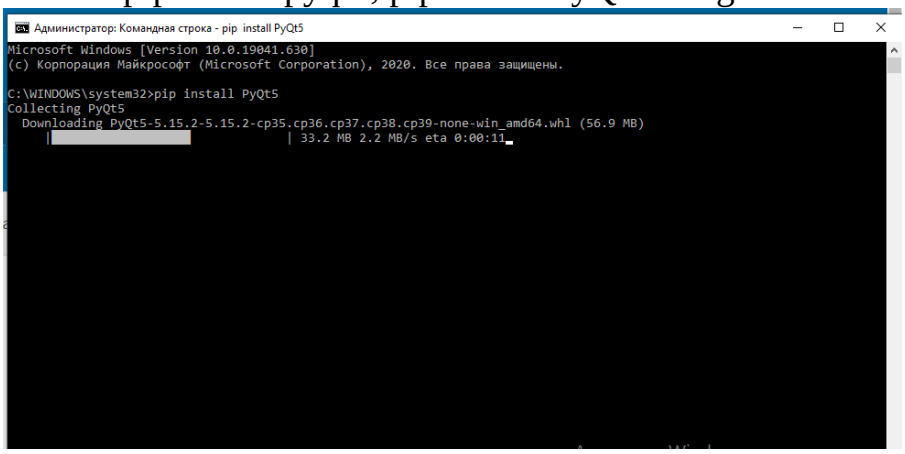

Рис.2. Инсталляция программы

После установки библиотек, запускаем дизайнер и создаем приложение.

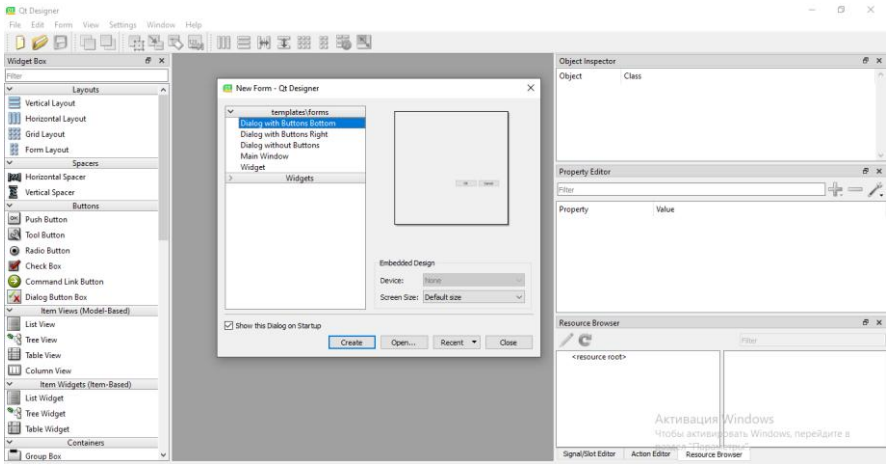

Рис. 3.Диалоговое окно Qt Designer

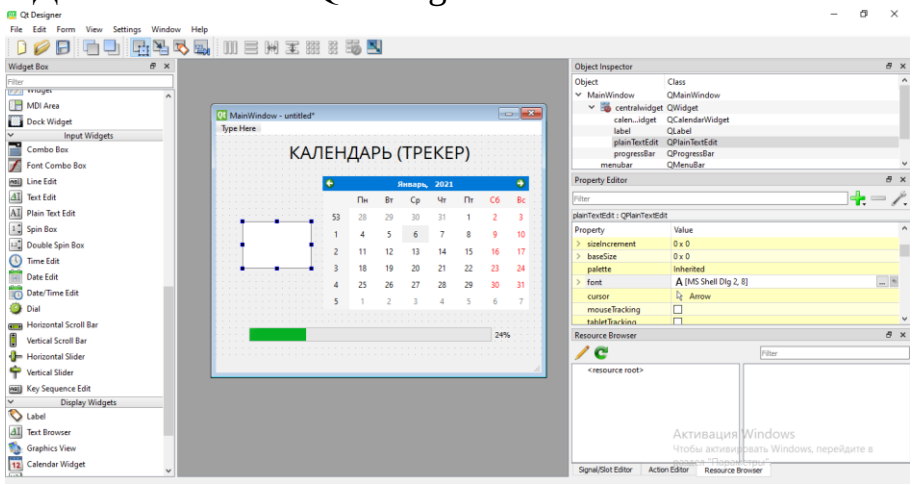

Рис. 4 Создание формы

Для создания окна формы описание какого то события, из панели инструментов DisplayWidgets, которая находится слева, в рабочую область перетаскиваются нужные элементы, это - Label, Calendar Widget, data Edit,

push Button, progress Bar. (рисунок 4). На панели инструментов с права, Property Editor можем изменять свойства событий. Устанавливать параметры.

Есть два способа запуска приложения:

1) Так как файл, созданный в Qt Designer сохраняется с расширением .ui, прежде чем его запустить, необходимо конвертировать файл .ui в файл с расширением .py. Конвертация осуществляется выполнением следующей команды в консоли: pyuic5 tracker.ui -o tracker.py . Затем сконвертированный файл импортируется в исполняемый код Python.
2) Загрузить файл .ui в исполняемый код Python, используя функцию loadUI(”tracker.ui”)

На рис.5. показана код tracker.ui находятся xml-файлы описание интерфейса в текстовом редакторе SublimeText.

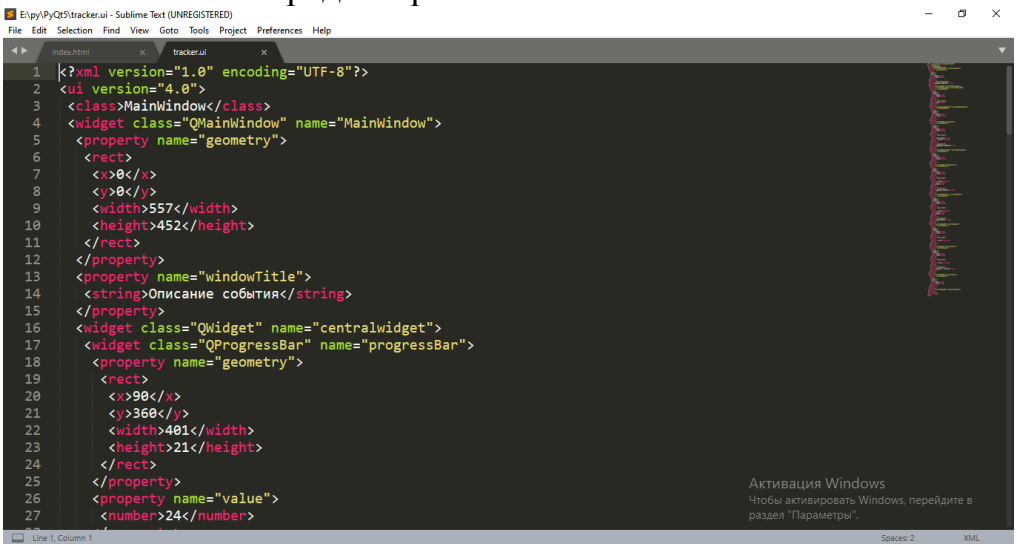

Рис.5. tracker.ui

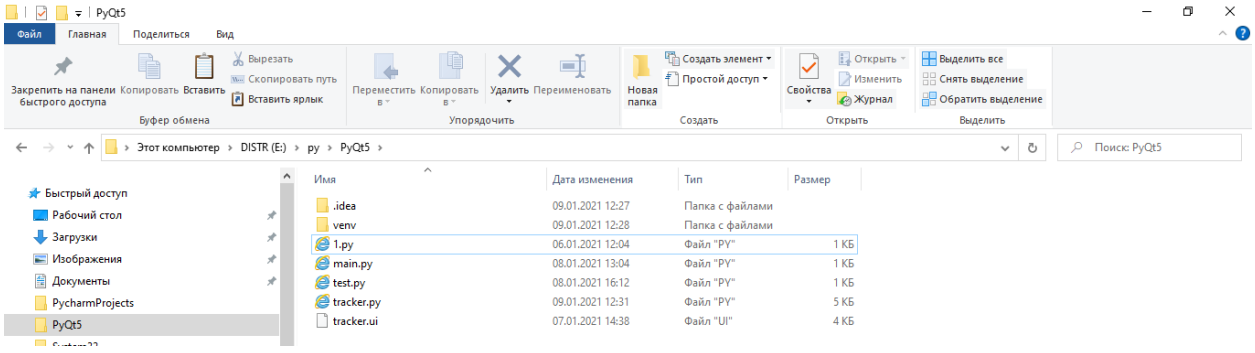

Рис. 6. Структура программы

После создания из Qt Designer, как tracker.ui, создаем другой файл на питоне.с помощью следующей программы

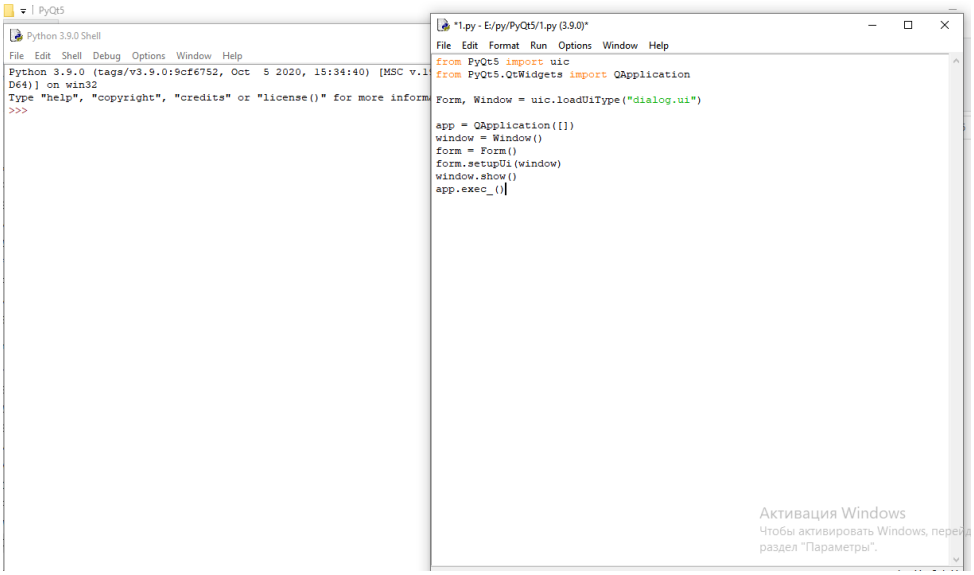

Рис.7.

```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication
Form, Window = uic.loadUiType("tracker.ui")
app = QApplication([])
window = Window()
form = Form()
form.setupUi(window)
window.show()
app.exec_()
```

С помощью кода мы активируем созданный интерфейс, и диалоговое окно должно выглядит следующим образом.

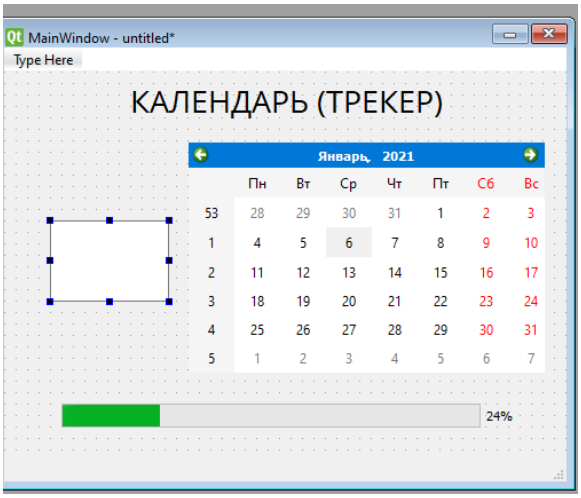

Рис. 8. Окно, созданное PyQt

Ниже приведен конвертированный из tracker.ui в tracker.py файл, где можно легко менять код, т.е. он представлен в виде объектно-ориентированного файла с использованием классов.

```
from PyQt5.QtWidgets import QApplocation, QtCore, QtGui, QtWidgets
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(557, 452)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.progressBar = QtWidgets.QProgressBar(self.centralwidget)
        self.progressBar.setGeometry(QtCore.QRect(90, 360, 401, 21))
        self.progressBar.setProperty("value", 24)
        self.progressBar.setObjectName("progressBar")
        self.calendarWidget = QtWidgets.QCalendarWidget(self.centralwidget)
        self.calendarWidget.setGeometry(QtCore.QRect(230, 80, 311, 191))
        self.calendarWidget.setObjectName("calendarWidget")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(120, 10, 301, 51))
        font = QtGui.QFont()
        font.setFamily("Open Sans")
        font.setPointSize(20)
        self.label.setFont(font)
        self.label.setObjectName("label")
        self.plainTextEdit = QtWidgets.QPlainTextEdit(self.centralwidget)
        self.plainTextEdit.setGeometry(QtCore.QRect(10, 120, 201, 121))
        self.plainTextEdit.setObjectName("plainTextEdit")
        self.label_2 = QtWidgets.QLabel(self.centralwidget)
        self.label_2.setGeometry(QtCore.QRect(20, 80, 171, 31))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.label_2.setFont(font)
        self.label_2.setObjectName("label_2")
        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(210, 310, 141, 31))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton.setFont(font)
        self.pushButton.setObjectName("pushButton")
        self.dateEdit = QtWidgets.QDateEdit(self.centralwidget)
        self.dateEdit.setGeometry(QtCore.QRect(50, 250, 110, 22))
        font = QtGui.QFont()
        font.setPointSize(11)
        self.dateEdit.setFont(font)
```

```
            self.dateEdit.setObjectName("dateEdit")
            self.label_3 = QtWidgets.QLabel(self.centralwidget)
            self.label_3.setGeometry(QtCore.QRect(220, 390, 161, 21))
            font = QtGui.QFont()
            font.setPointSize(12)
            self.label_3.setFont(font)
            self.label_3.setObjectName("label_3")
            MainWindow.setCentralWidget(self.centralwidget)
            self.menubar = QtWidgets.QMenuBar(MainWindow)
            self.menubar.setGeometry(QtCore.QRect(0, 0, 557, 21))
            self.menubar.setObjectName("menubar")
            MainWindow.setMenuBar(self.menubar)
            self.statusbar = QtWidgets.QStatusBar(MainWindow)
            self.statusbar.setObjectName("statusbar")
            MainWindow.setStatusBar(self.statusbar)
            self.retranslateUi(MainWindow)
            QtCore.QMetaObject.connectSlotsByName(MainWindow)
        def retranslateUi(self, MainWindow):
            _translate = QtCore.QCoreApplication.translate
            MainWindow.setWindowTitle(_translate("MainWindow", "Описание
события"))
            self.label.setText(_translate("MainWindow", "КАЛЕНДАРЬ
(ТРЕКЕР)"))
            self.label_2.setText(_translate("MainWindow", "Описание события"))
            self.pushButton.setText(_translate("MainWindow", "Отслеживать"))
            self.label_3.setText(_translate("MainWindow", "Осталось XX
дней"))
    if __name__ == "__main__":
        import sys
        app = QtWidgets.QApplication(sys.argv)
        MainWindow = QtWidgets.QMainWindow()
        ui = Ui_MainWindow()
        ui.setupUi(MainWindow)
        MainWindow.show()
        sys.exit(app.exec_())
```

В статье рассмотрена одна из самых популярных языков программирования Питон. Представлена история языка, основные библиотеки и среда разработки приложений PyQt5и QtDesigner. Виджеты PyQt5 поддерживает кнопки, метки, переключатели, поля и многое другое. С помощью QtDesigner можно сэкономить время на создание приложений, пользовательского интерфейса. Основываясь на предыдущий пример, можно сказать, что создание GUI приложения это помощь разработчикам в реализации своих идей.

Библиографический список

1. Голошубова О.М., Наумов В.Ю. Python. Происхождение, преимущества и перспективы. Инновационные, информационные и коммуникационные технологии. 2016. № 1. С. 38-40.
2. Админ. 2018 URL: https://proglib.io/p/python-applications
3. [б.а., б.н.] 2019. URL: https://habr.com/ru/post/481432/
4. Копытова М.А. Актуальность Языка Программирования Python. Экономика и социум. 2016. № 10 (29). С. 933-935.
5. Пальмов С.В. Сравнение библиотек tkInter, PyQt, wxPython. Евразийское Научное Объединение. 2020. № 11-2 (69). С. 122-126.
6. Москвитин А.А. Решение задач на компьютерах: часть II. Разработка программных средств: учебное пособие. М.-Берлин: Директ-Медиа, 2015. 427 с.
7. Joseph L., Learning Robotics Using Python. UK: Packt Publishing, 2018. 273c.
8. Шпагин О. Изучаем мир IT. https://wiseplat.org/infok. Москва.2020.
9. PyQt5 Tutorial: Design GUI using PyQt in Python with Examples. https://www.guru99.com/pyqt-tutorial.html
10. Hetland M. L. Beginning Python. From Novice to Professional. Second South Asian Edition, 2020.
11. Буйначев С.К., Боклаг Н.Ю. Основы программирования на языке Python. Учебное пособие для студентов. 2014.