МИНИСТЕРСТВО НАУКИ и высшего ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

«АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Колледж Алтайского государственного университета

Отделение экономики и информационных технологий

ОТЧЁТ ПО ПРАКТИКЕ

|  |  |
| --- | --- |
|  | Выполнил студент 3 курса 283б–сп группы Напалкова П.Н.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Научный руководитель Овсянников А.В.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Работа защищена  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Барнаул 2020

# ПОСТАНОВКА ЗАДАЧИ

Выполнить следующие пункты:

а. Создание базы данных с экзаменационными вопросами.

б. Приложение, поочередно выводящее экзаменационные вопросы, с конечным выводом количества успешных ответов

Тематика приложения это тестирование на знания ПДД.

# МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

2.1 Процессы в приложении

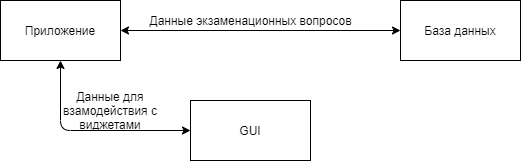
Для создания приложения была использована платформа PyQt5

При запуске приложения устанавливается соединение с базой данных. Затем, в случае успешного подключения в данные формы загружаются необходимые вопросы. В обработчике кнопки «начать тестирование» происходит вывод первого экзаменационного вопроса. В обработчике одной из трех кнопок выбора варианта ответа происходит сверка выбранного ответа с правильным ответом, полученного из базы данных. В обработчике кнопки «далее» происходит выбор следующего варианта ответа, полученного из базы данных. В обработчике кнопки «еще раз» происходит перезапуск тестирования т.е начало тестирования с первого вопроса.

2.2 Процессы в базе данных

База данных была выбрана MySQL, в которой была создана таблица с информацией по каждому экзаменационному вопросу.

# БЛОК СХЕМА



# Рисунок 1 – Блок схема

На блок схеме изображены все основные узлы данного проекта.

* Данные экзаменационных вопросов – поток данных, который возникает в процессе взаимодействия приложения с базой данных.
* Данные для взаимодействия с виджетами – поток данных, который возникает при взаимодействии программы и графического интерфейса.

# АЛГОРИТМ

Приложение взаимодействуя с базой данных, получает данные по экзаменационным вопросам. Далее приложение получив запрос, выдает поочередно экзаменационные вопросы, при этом запоминая варианты ответа пользователя. Как итог приложение выдает количество правильных ответов.

# ЗАКЛЮЧЕНИЕ

В ходе практической работы была создана связка приложение – база данных, отлажена их работа, наполнена база данных.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Маркин, А.В. Основы web-программирования на PHP: учебное пособие [Электронный ресурс] / А.В. Маркин, С.С. Шкарин. - Москва: Диалог-МИФИ, 2012. - 252 с. - доступ из ЭБС «Университетская библиотека он-лайн»

## ПРИЛОЖЕНИЕ А

## Исходный код приложения:

|  |
| --- |
| import MySQLdb |
|  | import sys |
|  | from PyQt5 import uic, QtGui |
|  | from PyQt5.QtWidgets import \* |
|  | from PyQt5.QtGui import QPixmap |
|  |  |
|  | # функция, которая вызывается при нажатии кнопки "показать результат" |
|  | def finish(): |
|  | window.manage\_but.clicked.disconnect() |
|  | window.manage\_but.clicked.connect(getDatabase) |
|  | global historyAns |
|  | global currentQuestion |
|  | window.photoLabel.setVisible(False) |
|  | window.question\_label.setVisible(False) |
|  | window.first\_answer.setVisible(False) |
|  | window.second\_answer.setVisible(False) |
|  | window.third\_answer.setVisible(False) |
|  | window.manage\_but.setVisible(False) |
|  | window.explain\_lable.setVisible(False) |
|  | window.onceAgain.setVisible(True) |
|  | window.finishLabel.setText("Количество правильных ответов = " + str(round(historyAns/(lenRows + 1) \* 100)) + "%") |
|  | # функция, которая вызывается при нажатии "еще раз" для отображения нужных элементов |
|  | def startAgain(): |
|  | global currentQuestion |
|  | global historyAns |
|  | window.finishLabel.clear() |
|  | window.photoLabel.setVisible(True) |
|  | window.question\_label.setVisible(True) |
|  | window.first\_answer.setVisible(True) |
|  | window.second\_answer.setVisible(True) |
|  | window.third\_answer.setVisible(True) |
|  | window.manage\_but.setVisible(True) |
|  | window.explain\_lable.setVisible(True) |
|  | window.onceAgain.setVisible(False) |
|  |  |
|  | currentQuestion = 0 |
|  | historyAns = 0 |
|  | getDatabase() |
|  | # функция, которая вызвается при нажатии на любого вариант ответа для его проверки и занесения ответа в массив |
|  | def checkAnswer(numberButton, objectsButton): |
|  | global currentQuestion |
|  | global numberRightAns |
|  | global lenRows |
|  | global historyAns |
|  | right\_ans = rows[currentQuestion]['right\_answer'] |
|  | window.explain\_lable.setText(rows[currentQuestion]['explain']) |
|  | window.manage\_but.setEnabled(True) |
|  | window.first\_answer.setEnabled(False) |
|  | window.second\_answer.setEnabled(False) |
|  | window.third\_answer.setEnabled(False) |
|  |  |
|  | if (right\_ans == numberButton): |
|  | objectsButton[numberButton].setStyleSheet("background-color: green; color: white;") |
|  | historyAns += 1 |
|  | else: |
|  | objectsButton[numberButton].setStyleSheet("background-color: red; color: white;") |
|  | objectsButton[right\_ans].setStyleSheet("background-color: green; color: white;") |
|  |  |
|  | if (currentQuestion == lenRows): |
|  | window.manage\_but.setText("Показать результат") |
|  | window.manage\_but.clicked.connect(finish) |
|  | else: |
|  | currentQuestion += 1 |
|  | # функция, которая вызывается при нажатии кнопки "далее" для получения следующего вопроса из базы данных |
|  | def getDatabase(): |
|  | global currentQuestion |
|  | window.first\_answer.setEnabled(True) |
|  | window.second\_answer.setEnabled(True) |
|  | window.third\_answer.setEnabled(True) |
|  | window.first\_answer.setStyleSheet("") |
|  | window.second\_answer.setStyleSheet("") |
|  | window.third\_answer.setStyleSheet("") |
|  | window.explain\_lable.clear() |
|  | window.first\_answer.setText(rows[currentQuestion]['first\_answer']) |
|  | window.second\_answer.setText(rows[currentQuestion]['second\_answer']) |
|  | window.third\_answer.setText(rows[currentQuestion]['third\_answer']) |
|  | window.question\_label.setText(rows[currentQuestion]['question']) |
|  | window.manage\_but.setText("Далее") |
|  | window.manage\_but.setEnabled(False) |
|  |  |
|  | pixmap = QPixmap("pictures/" + rows[currentQuestion]['picture\_name']) |
|  | pixmap = pixmap.scaledToWidth(780) |
|  | window.photoLabel.setPixmap(pixmap) |
|  |  |
|  |  |
|  | class MainWindow(QMainWindow): |
|  | def \_\_init\_\_(self): |
|  | super(MainWindow, self).\_\_init\_\_() |
|  | uic.loadUi('gui.ui', self) |
|  | self.setWindowTitle('Проверка знаний ПДД') |
|  |  |
|  | if \_\_name\_\_ == '\_\_main\_\_': |
|  | app = QApplication(sys.argv) # создание объекта самого приложения |
|  | app.setWindowIcon(QtGui.QIcon('pictures/icon.png')) |
|  | window = MainWindow() |
|  |  |
|  | global currentQuestion |
|  | global lenRows |
|  | global historyAns |
|  | historyAns = 0 |
|  | currentQuestion = 0 |
|  |  |
|  | conn = MySQLdb.connect('localhost', 'root', 'root', 'pdd', charset='utf8') |
|  | cursor = conn.cursor(MySQLdb.cursors.DictCursor) |
|  | cursor.execute("SELECT \* FROM pdd.questions;") |
|  | rows = cursor.fetchall() |
|  | lenRows = len(rows) - 1 |
|  |  |
|  | objectsButton = [0] \* 4 |
|  | objectsButton[1] = window.first\_answer |
|  | objectsButton[2] = window.second\_answer |
|  | objectsButton[3] = window.third\_answer |
|  |  |
|  | window.first\_answer.clicked.connect(lambda: checkAnswer(1, objectsButton)) |
|  | window.second\_answer.clicked.connect(lambda: checkAnswer(2, objectsButton)) |
|  | window.third\_answer.clicked.connect(lambda: checkAnswer(3, objectsButton)) |
|  | window.onceAgain.clicked.connect(startAgain) |
|  | window.manage\_but.clicked.connect(getDatabase) |
|  |  |
|  | # в начале сделать кнопки некликабельными |
|  | window.onceAgain.setVisible(False) |
|  | window.first\_answer.setEnabled(False) |
|  | window.second\_answer.setEnabled(False) |
|  | window.third\_answer.setEnabled(False) |
|  | window.manage\_but.setText("Начать тестирование") |
|  | # показать настроенное окно GUI |
|  | window.show() |
|  | # завершение работы программы |
|  | cursor.close() |
|  | sys.exit(app.exec\_()) |

Код графического интерфейса (gui.ui):

from PySide2.QtCore import \*

from PySide2.QtGui import \*

from PySide2.QtWidgets import \*

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

if not MainWindow.objectName():

MainWindow.setObjectName(u"MainWindow")

MainWindow.resize(800, 750)

sizePolicy = QSizePolicy(QSizePolicy.Ignored, QSizePolicy.Ignored)

sizePolicy.setHorizontalStretch(0)

sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightForWidth())

MainWindow.setSizePolicy(sizePolicy)

MainWindow.setMinimumSize(QSize(800, 750))

MainWindow.setMaximumSize(QSize(800, 750))

self.centralwidget = QWidget(MainWindow)

self.centralwidget.setObjectName(u"centralwidget")

self.first\_answer = QPushButton(self.centralwidget)

self.first\_answer.setObjectName(u"first\_answer")

self.first\_answer.setGeometry(QRect(25, 420, 750, 50))

self.first\_answer.setMinimumSize(QSize(750, 0))

self.first\_answer.setMaximumSize(QSize(750, 16777215))

font = QFont()

font.setPointSize(9)

font.setBold(False)

font.setWeight(50)

self.first\_answer.setFont(font)

self.third\_answer = QPushButton(self.centralwidget)

self.third\_answer.setObjectName(u"third\_answer")

self.third\_answer.setGeometry(QRect(25, 518, 750, 50))

self.third\_answer.setFont(font)

self.second\_answer = QPushButton(self.centralwidget)

self.second\_answer.setObjectName(u"second\_answer")

self.second\_answer.setGeometry(QRect(25, 469, 750, 50))

self.second\_answer.setFont(font)

self.explain\_lable = QLabel(self.centralwidget)

self.explain\_lable.setObjectName(u"explain\_lable")

self.explain\_lable.setGeometry(QRect(30, 631, 741, 81))

font1 = QFont()

font1.setPointSize(11)

self.explain\_lable.setFont(font1)

self.explain\_lable.setLayoutDirection(Qt.LeftToRight)

self.explain\_lable.setFrameShape(QFrame.NoFrame)

self.explain\_lable.setAlignment(Qt.AlignCenter)

self.explain\_lable.setWordWrap(True)

self.photoLabel = QLabel(self.centralwidget)

self.photoLabel.setObjectName(u"photoLabel")

self.photoLabel.setGeometry(QRect(10, 10, 780, 320))

self.manage\_but = QPushButton(self.centralwidget)

self.manage\_but.setObjectName(u"manage\_but")

self.manage\_but.setGeometry(QRect(25, 578, 750, 40))

font2 = QFont()

font2.setPointSize(12)

self.manage\_but.setFont(font2)

self.question\_label = QLabel(self.centralwidget)

self.question\_label.setObjectName(u"question\_label")

self.question\_label.setGeometry(QRect(30, 340, 741, 71))

font3 = QFont()

font3.setPointSize(12)

font3.setBold(True)

font3.setWeight(75)

self.question\_label.setFont(font3)

self.question\_label.setAlignment(Qt.AlignCenter)

self.question\_label.setWordWrap(True)

self.onceAgain = QPushButton(self.centralwidget)

self.onceAgain.setObjectName(u"onceAgain")

self.onceAgain.setGeometry(QRect(225, 571, 356, 70))

font4 = QFont()

font4.setPointSize(19)

self.onceAgain.setFont(font4)

self.finishLabel = QLabel(self.centralwidget)

self.finishLabel.setObjectName(u"finishLabel")

self.finishLabel.setGeometry(QRect(20, 70, 751, 271))

font5 = QFont()

font5.setPointSize(36)

self.finishLabel.setFont(font5)

self.finishLabel.setAlignment(Qt.AlignCenter)

self.finishLabel.setWordWrap(True)

MainWindow.setCentralWidget(self.centralwidget)

self.menubar = QMenuBar(MainWindow)

self.menubar.setObjectName(u"menubar")

self.menubar.setGeometry(QRect(0, 0, 800, 26))

MainWindow.setMenuBar(self.menubar)

self.statusbar = QStatusBar(MainWindow)

self.statusbar.setObjectName(u"statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

QMetaObject.connectSlotsByName(MainWindow)

# setupUi

def retranslateUi(self, MainWindow):

MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow", u"MainWindow", None))

self.first\_answer.setText("")

self.third\_answer.setText("")

self.second\_answer.setText("")

self.explain\_lable.setText("")

self.photoLabel.setText("")

self.manage\_but.setText(QCoreApplication.translate("MainWindow", u"\u0414\u0430\u043b\u0435\u0435", None))

self.question\_label.setText("")

self.onceAgain.setText(QCoreApplication.translate("MainWindow", u"\u0415\u0449\u0435 \u0440\u0430\u0437", None))

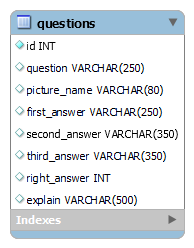
self.finishLabel.setText("")

# retranslateUi

## ПРИЛОЖЕНИЕ В:

Рисунок 2 – Страница товаров

Рисунок 3 – Страница товара



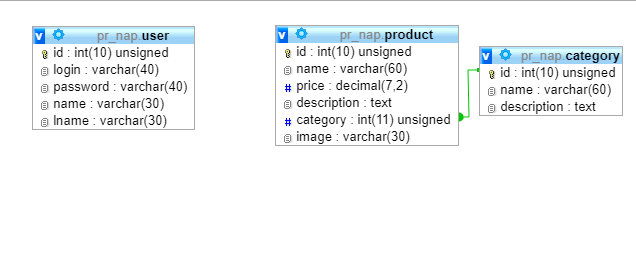


Рисунок 5 – Структура базы данных