

Практическое занятие № 15

Тема: Составление программ в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community.

Постановка задачи №1.

Приложение РАСХОДЫ ПО ВИДАМ ПРОДУКЦИИ для автоматизированного контроля затрат на производство продукции. БД должна содержать таблицу Расходы со следующей структурой записи: Дата, Код продукта, Наименование продукта, Расходы, Сумма. БД должна обеспечивать получение информации о расходах на производство продукции по наименованию продукта.

Текст программы:

```
from tkinter import *  
import sqlite3 as sq
```

```
def open_update_dialog():  
    Update()
```

```
def open_search_dialog():  
    Search()
```

```
def open_dialog():  
    Child(root, app)
```

```
class Main(Frame):  
    def __init__(self, root):  
        super().__init__(root)  
        self.tree = None  
        self.refresh_img = None  
        self.search_img = None  
        self.delete_img = None  
        self.update_img = None  
        self.btn_open_dialog = None  
        self.add_img = None  
        self.init_main()  
        self.db = db  
        self.view_records()
```

```
def init_main(self):
```

```

toolbar = Frame(bg='#00a8f3', bd=4)
toolbar.pack(side=TOP, fill=X)

self.add_img = PhotoImage(file="11.gif")
self.btn_open_dialog = Button(toolbar, text='Добавить продукт',
command=open_dialog, bg='#00a8f3',
bd=0, compound=TOP, image=self.add_img)
self.btn_open_dialog.pack(side=LEFT)

self.update_img = PhotoImage(file="12.gif")
btn_edit_dialog = Button(toolbar, text="Редактировать",
command=open_update_dialog, bg='#00a8f3',
bd=0, compound=TOP, image=self.update_img)
btn_edit_dialog.pack(side=LEFT)

self.delete_img = PhotoImage(file="13.gif")
btn_delete = Button(toolbar, text="Удалить запись", command=self.delete_records,
bg='#00a8f3',
bd=0, compound=TOP, image=self.delete_img)
btn_delete.pack(side=LEFT)

self.search_img = PhotoImage(file="14.gif")
btn_search = Button(toolbar, text="Поиск записи", command=open_search_dialog,
bg='#00a8f3',
bd=0, compound=TOP, image=self.search_img)
btn_search.pack(side=LEFT)

self.refresh_img = PhotoImage(file="15.gif")
btn_refresh = Button(toolbar, text="Обновить экран", command=self.view_records,
bg='#00a8f3',
bd=0, compound=TOP, image=self.refresh_img)
btn_refresh.pack(side=LEFT)

from tkinter.ttk import Treeview
self.tree = Treeview(self, columns=('data', 'code', 'name', 'expenses', 'amount'),
height=15, show='headings')

self.tree.column('data', width=120, anchor=CENTER)
self.tree.column('code', width=140, anchor=CENTER)
self.tree.column('name', width=220, anchor=CENTER)
self.tree.column('expenses', width=140, anchor=CENTER)
self.tree.column('amount', width=160, anchor=CENTER)

self.tree.heading('data', text='Дата')
self.tree.heading('code', text='Код продукта')
self.tree.heading('name', text='Наименование продукта')
self.tree.heading('expenses', text='Расходы')

```

```

self.tree.heading('amount', text='Сумма')

self.tree.pack()

def records(self, data, code, name, expenses, amount):
    self.db.insert_data(data, code, name, expenses, amount)
    self.view_records()

def update_record(self, data, code, name, expenses, amount):
    self.db.cur.execute("""UPDATE susers SET data=?, code=?, name=?, expenses=?,
amount=?
WHERE user_id=?""",
                        (data, code, name, expenses, amount,
                         self.tree.set(self.tree.selection()[0], '#1')))
    self.db.con.commit()
    self.view_records()

def view_records(self):
    self.db.cur.execute("""SELECT * FROM susers""")
    [self.tree.delete(i) for i in self.tree.get_children()]
    [self.tree.insert("", 'end', values=row) for row in self.db.cur.fetchall()]

def delete_records(self):
    for selection_item in self.tree.selection():
        self.db.cur.execute("""DELETE FROM susers WHERE data=?""",
        (self.tree.set(selection_item, '#1'),))
        self.db.con.commit()
        self.view_records()

def search_records(self, title):
    title = (title,)
    self.db.cur.execute("""SELECT * FROM susers WHERE name>=?""", title)
    [self.tree.delete(i) for i in self.tree.get_children()]
    [self.tree.insert("", 'end', values=row) for row in self.db.cur.fetchall()]

class Child(Toplevel):

    def __init__(self, root, app):
        super().__init__(root)
        self.btn_ok = None
        self.entry_amount = None
        self.entry_expenses = None
        self.entry_name = None
        self.entry_code = None
        self.entry_data = None
        self.init_child()

```

```

self.view = app

def init_child(self):
    self.title('Добавить препарат')
    self.geometry('400x250')
    self.resizable(False, False)

    label_data = Label(self, text='Дана')
    label_data.place(x=30, y=25)
    self.entry_data = Entry(self)
    self.entry_data.place(x=180, y=25)

    label_code = Label(self, text='Код продукта')
    label_code.place(x=30, y=50)
    self.entry_code = Entry(self)
    self.entry_code.place(x=180, y=50)

    label_name = Label(self, text='Наименование продукта')
    label_name.place(x=30, y=75)
    self.entry_name = Entry(self)
    self.entry_name.place(x=180, y=75)

    label_expenses = Label(self, text='Расходы')
    label_expenses.place(x=30, y=100)
    self.entry_expenses = Entry(self)
    self.entry_expenses.place(x=180, y=100)

    label_amount = Label(self, text='Сумма')
    label_amount.place(x=30, y=125)
    self.entry_amount = Entry(self)
    self.entry_amount.place(x=180, y=125)

    btn_cancel = Button(self, text='Закрыть', command=self.destroy)
    btn_cancel.place(x=300, y=220)

    self.btn_ok = Button(self, text='Добавить')
    self.btn_ok.place(x=220, y=220)
    self.btn_ok.bind('<Button-1>', lambda event: self.view.records(self.entry_data.get(),
                                                                    self.entry_code.get(),
                                                                    self.entry_name.get(),
                                                                    self.entry_expenses.get(),
                                                                    self.entry_amount.get()))

    self.grab_set()
    self.focus_set()

```

```

class Update(Child):
    def __init__(self):
        super().__init__(root, app)
        self.init_edit()
        self.view = app

    def init_edit(self):
        self.title("Редактировать запись")
        btn_edit = Button(self, text="Редактировать")
        btn_edit.place(x=205, y=220)
        btn_edit.bind('<Button-1>', lambda event:
self.view.update_record(self.entry_data.get(),
                        self.entry_code.get(),
                        self.entry_name.get(),
                        self.entry_expenses.get(),
                        self.entry_amount.get()))

        self.btn_ok.destroy()

class Search(Toplevel):
    def __init__(self):
        super().__init__()
        self.entry_search = None
        self.init_search()
        self.view = app

    def init_search(self):
        self.title("Поиск")
        self.geometry("300x100+400+300")
        self.resizable(False, False)

        label_search = Label(self, text="Поиск")
        label_search.place(x=50, y=20)

        self.entry_search = Entry(self)
        self.entry_search.place(x=105, y=20, width=150)

        btn_cancel = Button(self, text="Заккрыть", command=self.destroy)
        btn_cancel.place(x=185, y=50)

        btn_search = Button(self, text="Поиск")
        btn_search.place(x=105, y=50)
        btn_search.bind('<Button-1>', lambda event:
self.view.search_records(self.entry_search.get()))
        btn_search.bind('<Button-1>', lambda event: self.destroy(), add='+')

```

```

class DB:
    def __init__(self):

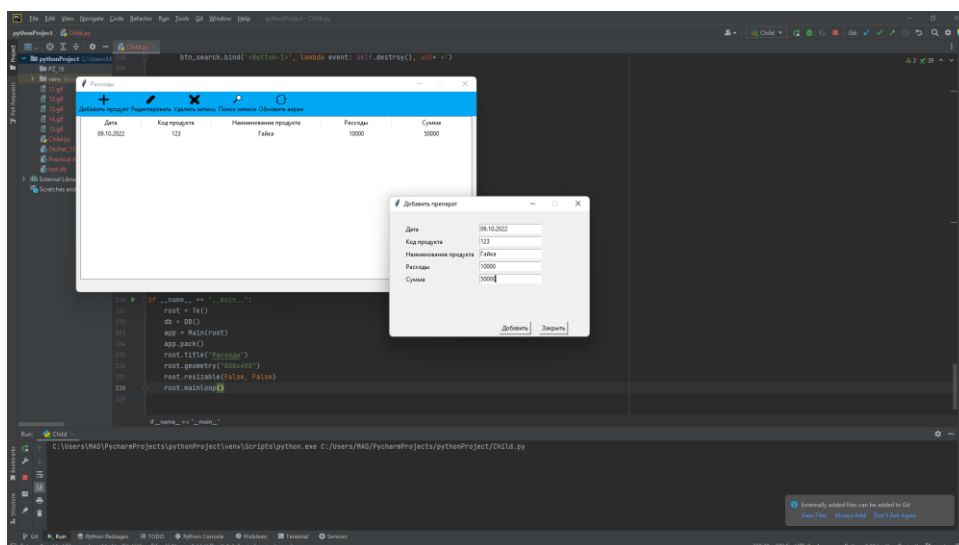
        with sq.connect('test.db') as self.con:
            self.cur = self.con.cursor()
            self.cur.execute("""CREATE TABLE IF NOT EXISTS susers (
                data INTEGER,
                code INTEGER,
                name TEXT NOT NULL,
                expenses INTEGER NOT NULL DEFAULT 1,
                amount INTEGER
            )""")

        def insert_data(self, data, code, name, expenses, amount):
            self.cur.execute("""INSERT INTO susers(data, code, name, expenses, amount)
            VALUES (?, ?, ?, ?, ?)""", (data, code, name, expenses, amount))
            self.con.commit()

if __name__ == "__main__":
    root = Tk()
    db = DB()
    app = Main(root)
    app.pack()
    root.title("Расходы")
    root.geometry("800x400")
    root.resizable(False, False)
    root.mainloop()

```

Протокол работы программы:



Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на *GitHub*.