



GuiPy – eine didaktische Python IDE

im Einsatz mit der Game Engine Graphics'n'Games

(Matthias Haupt – LMU und Erasmus-Grasser-Gymnasium München)



Gliederung

1. Einführung
2. Installation und Konfiguration
3. Praxis (Teil 1)
4. Ausgewählte Funktionen von GuiPy
5. Praxis (Teil 2)
6. Werkzeugvergleich & Diskussion



Einführung – Lehrplan (GY)

- 9. Jgstf.: Lernbereich 3: Grundlagen der objektorientierten Modellierung und Programmierung
- 10. Jgstf.:
 - Objektorientierte Modellierung und Programmierung (ca. 27 Std.)
 - Projekt (ca. 12 Std.)
- 11. Jgstf.
 - Graphen (ca. 10 Std.)
 - Kommunikation in Netzwerken, Internet (ca. 10 Std.)
 - Algorithmik (nicht NTG)
- 12. Jgstf.
- 13. Jgstf.



Einführung – Lehrplan (RS)

- **Modul 2.6.1. Modellieren und Codieren von Algorithmen**

„schreiben auf Basis zuvor erstellter Modelle Quellcode für Programme in einer **Entwicklungsumgebung**.

arbeiten mit Variablen, Datentypen, Funktionen und Kommentaren, um Programmcode übersichtlich und überprüfbar zu machen.[...]“

Inhalte zu den Kompetenzen:

„Notationsformen, z. B. **Struktogramm**, Pseudocode, Programmablaufplan

Quellcode: Syntax und Semantik der verwendeten Programmiersprache

Entwicklungsumgebung: Aufgaben (z. B. Syntaxhervorhebung, Kompilierung, Debugging) und Funktionsweise der verwendeten Umgebung

[...]“

- **Modul 2.6.2 Objektorientierte Softwareentwicklung**

„[...] implementieren objektorientierte Programme mit einem geeigneten Programmierwerkzeug.

testen und optimieren ihre objektorientierten Programme.

Inhalte zu den Kompetenzen:

standardisierte Notationsformen, z. B. **Klassendiagramm**, Programmablaufplan, **Struktogramm**, Aktivitätsdiagramm

[...]“



Einführung – Was ist GuiPy?

- Python-IDE für die Schule, basierend auf PyScripter
- vorgestellt 2022
- Entwickelt von Gerhard Röhner
- Bietet mehrere grafische Tools zur Unterstützung des Lernprozesses
- Weitere Informationen unter:

<https://gui.py.de/>



Installation und Konfiguration

- Installation Python

⇒ <https://www.python.org/downloads/>

- Installation pygame library

⇒ Cmd

⇒ pip3 install pygame

- Download GuiPy

=> <https://guipy.de/doku.php?id=en:download>

(ggf. als portable Version auf USB-Stick)



Installation und Konfiguration

- Verwendung der Game Engine Graphics'n'Games
 - Aus Buchreihe des Cornelsen-Verlags (www.informatikschulbuch.de)
 - Speziell angepasst an die Bedürfnisse des LehrplanPLUS Gym
- Für jede Aufgabe gibt es Vorlagen auf der Webseite
 - anwendung.py – eigentliche Aufgabenvorlage
 - graphics_and_games_klassen – enthält Oberklassen der Game Engine
 - Im Ordner intern: zeichenfenster – interne Programmierung des Zeichenfensters



Installation und Konfiguration

- Empfehlung Arbeit mit den Vorlagen
 - graphics_and_games_klassen und intern Ordner einmal entpacken
 - In **gleichen** Ordner die anwendung.py Dateien hineinkopieren (mit entsprechender Benennung)

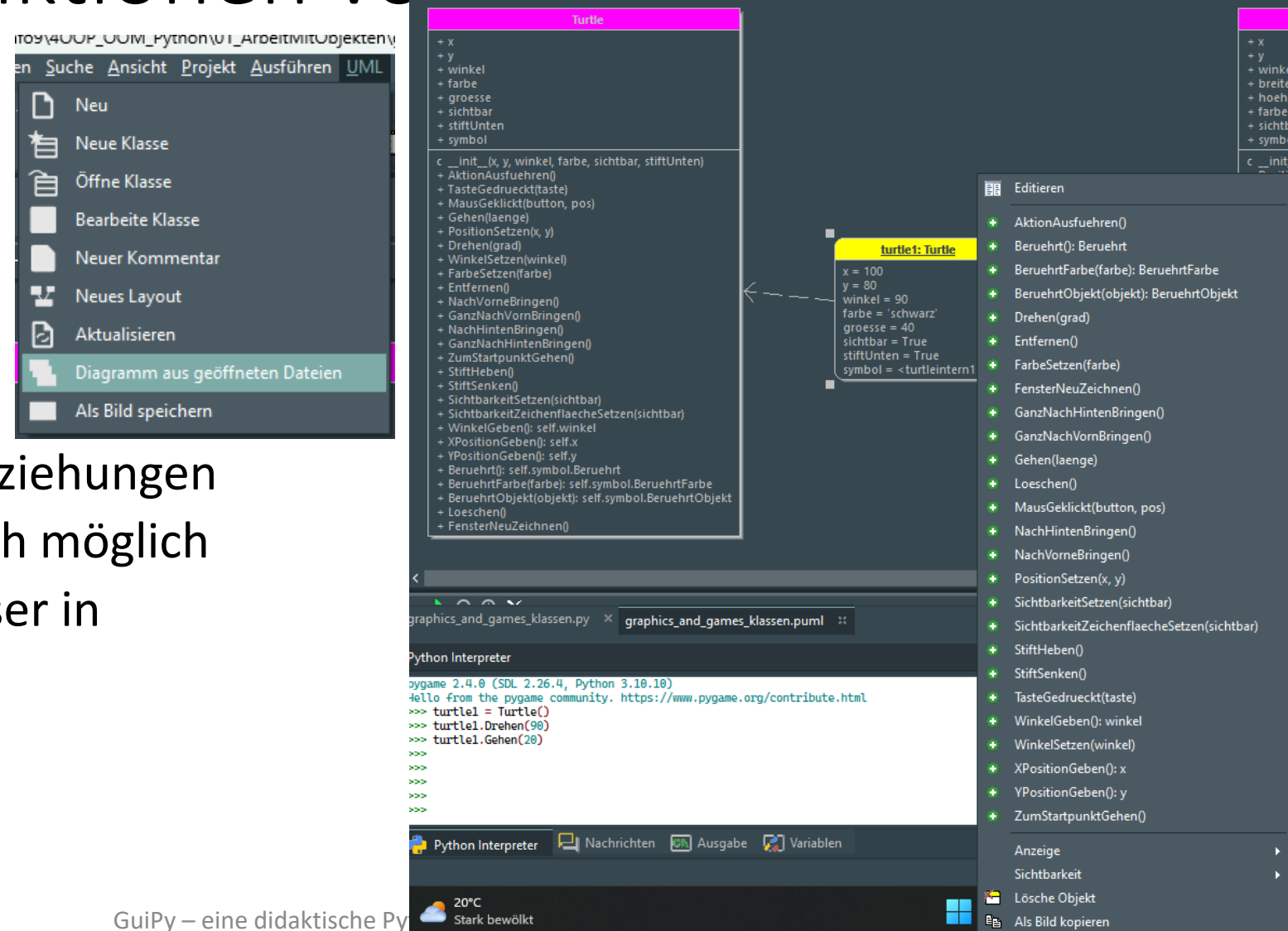
 intern	02.07.2023 15:27	Dateiordner	
 02_S_112_Erkundungsauftrag_Vorlage	17.03.2023 15:42	PY-Datei	1 KB
 03_S_118_Rasenmaeher	17.04.2023 12:44	PY-Datei	6 KB
 04 S 126 1	21.04.2023 10:57	PY-Datei	3 KB
 graphics_and_games_klassen	02.07.2023 13:39	PY-Datei	38 KB



Ausgewählte Funktionen von GuiPy

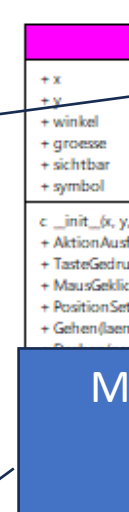
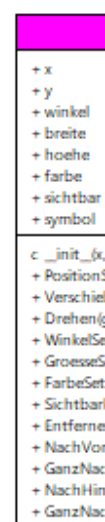
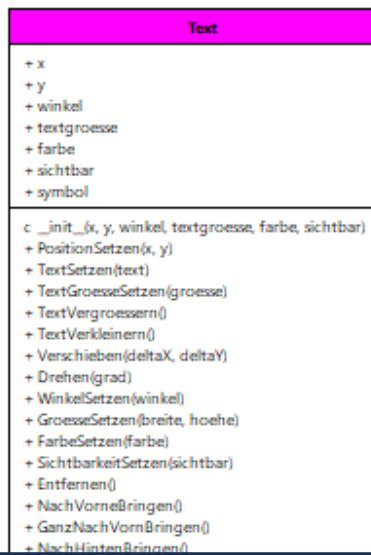
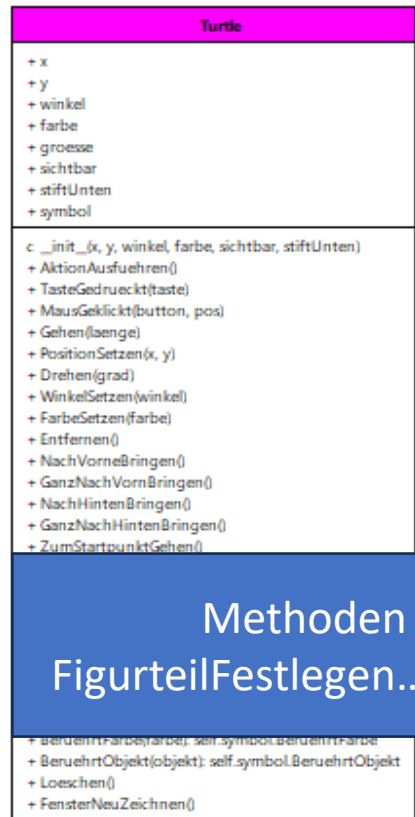
- Klassendiagramme und Objektdiagramme:

- + Zeigt auch Vererbungsbeziehungen
- + Methodenaufrufe grafisch möglich
- + Direkte Übersetzung dieser in Punktschreibweise
- + Aktualisierung der Attributwerte





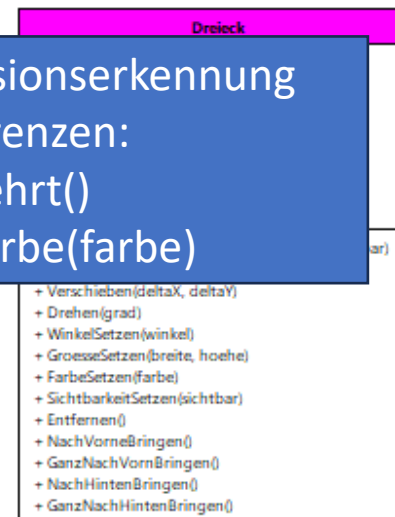
Arbeit mit Graphics'n'Games



Methode AktionAusfuehren():
Wird Takt für Takt aufgerufen

Methoden für Kollisionserkennung
ohne Referenzen:

- Beruehrt()
- BeruehrtFarbe(farbe)



Methoden zur Darstellung:
FigurteilFestlegen...(x,y,breite,höhe, farbe)

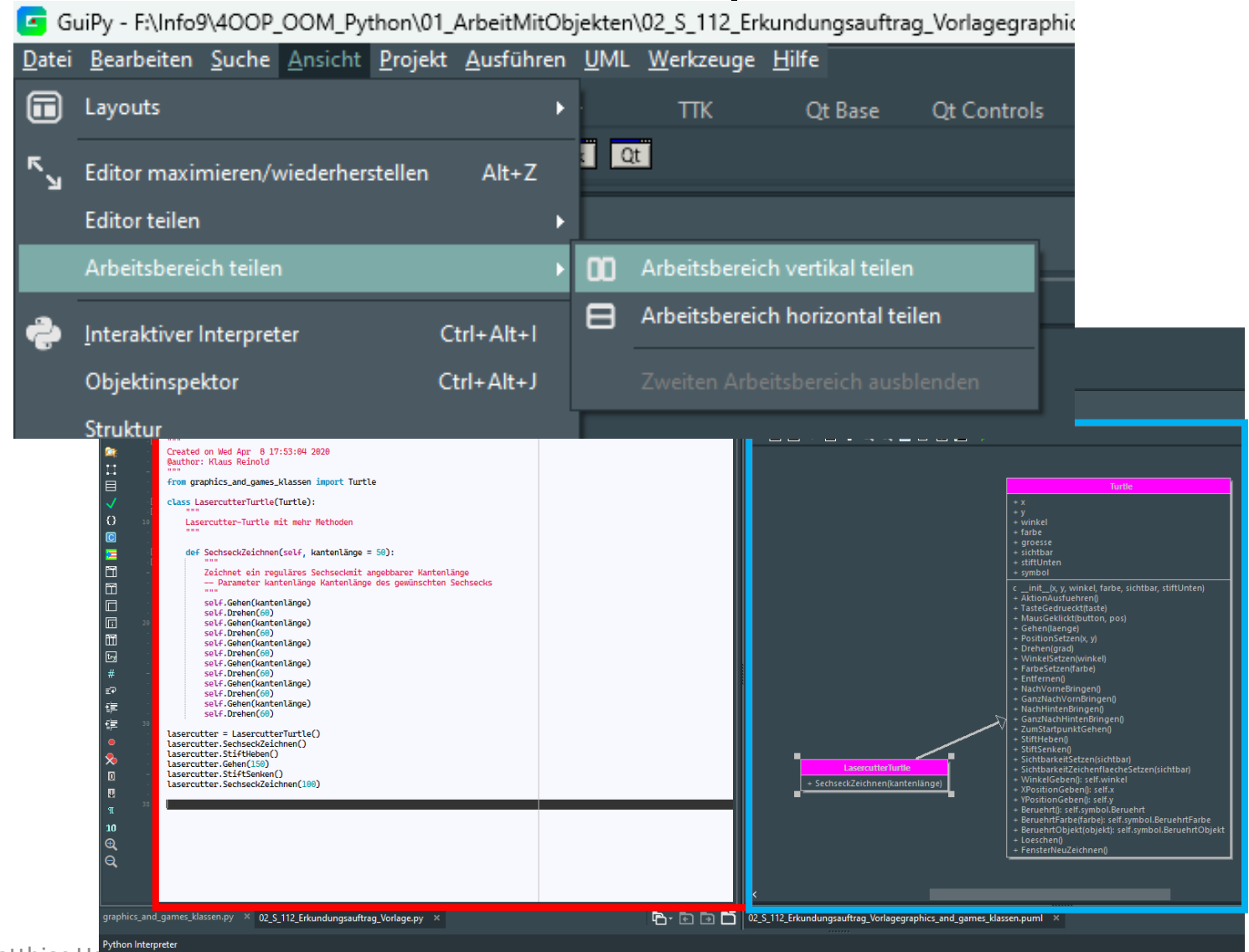


Ausgewählte Funktionen von GuiPy

Komfortables Arbeiten mit
Editor und
Klassendiagramm

+ Verfügbare Methoden
der Oberklasse direkt
ersichtlich

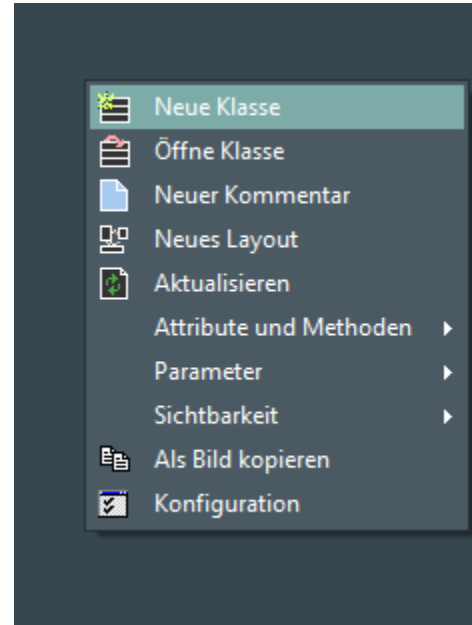
- Werden neue Methoden
geschrieben, muss
Diagramm neu erzeugt
werden
- Bei Nutzung der
Windows-
Bildschirmaufteilung
nicht mehr alles sichtbar



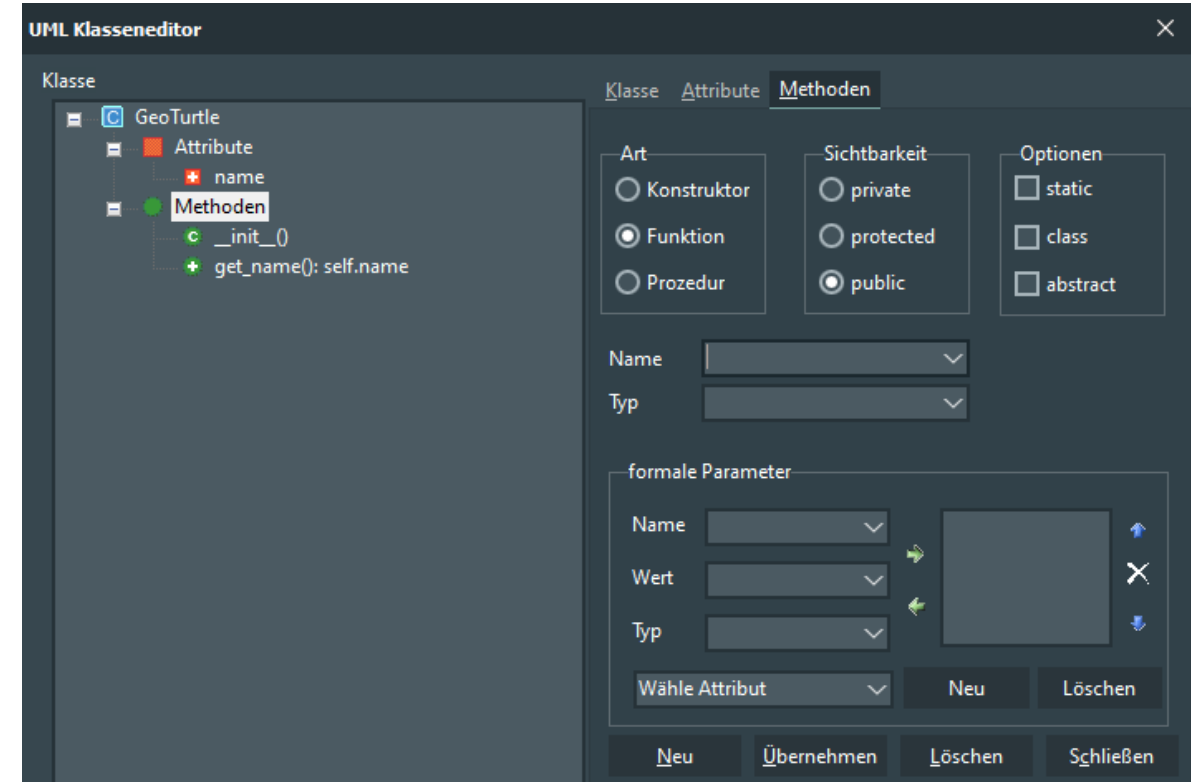


Ausgewählte Funktionen von GuiPy

+ UML <-> Code



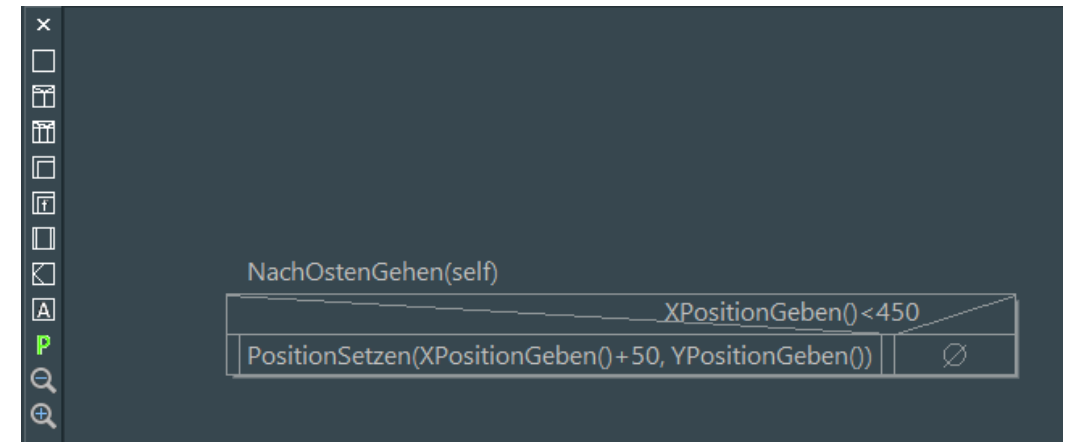
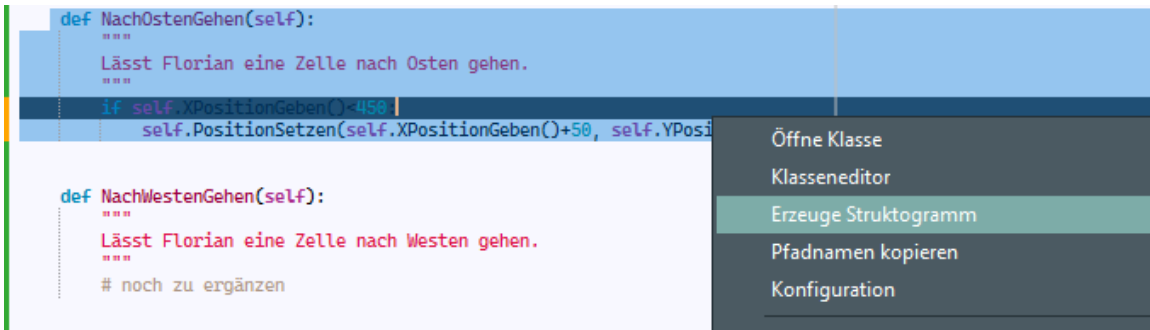
Grafisches
Editormenü aus
Quelltext heraus





Ausgewählte Funktionen von GuiPy

+ Struktogramm <-> Code



Editieren des
Struktogramms
möglich

Ausgewählte Funktionen von GuiPy

Leiste mit vielen Programmierhilfen:

+ Stride-ähnliches Arbeiten durch vorgegebene Struktur von Kontrollstrukturen/Syntax möglich

+ Einrückungshilfe

+ Klammerfindungs-Assistent

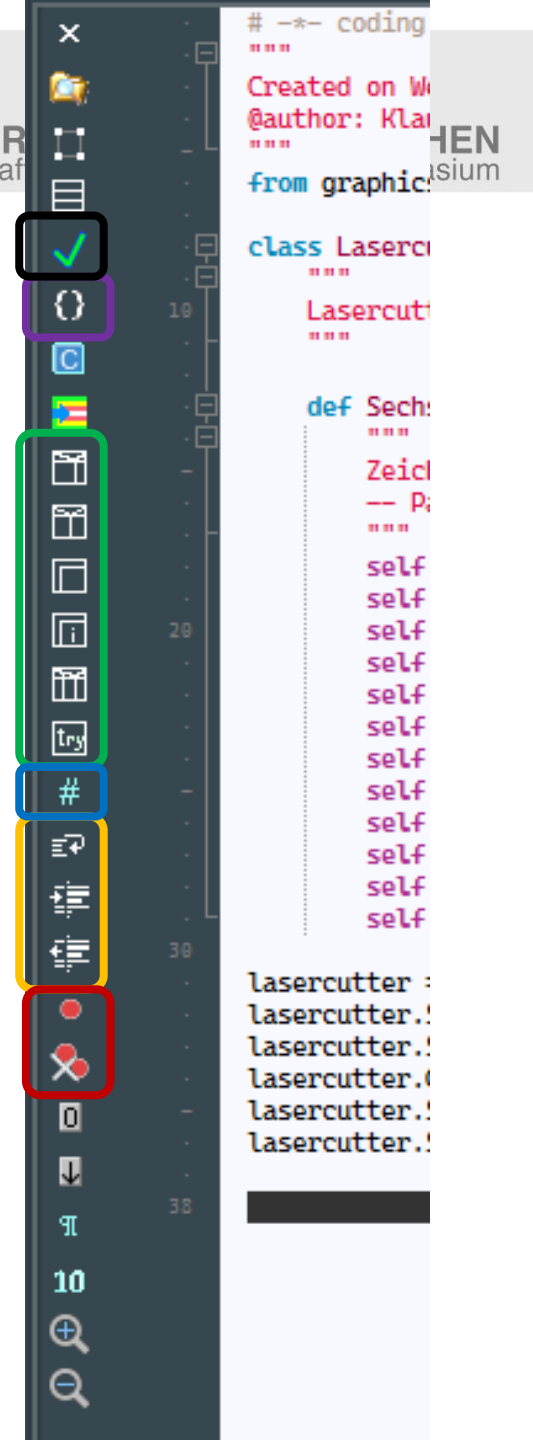
+ Aus/Einkommentiermöglichkeit

o Debugger (Funktion??)

+ Syntaxcheck

o weniger Quelltextkenntnis nötig?

- ggf. unübersichtlich (Leiste kann nicht entfernt werden)





Ausgewählte Funktionen von GuiPy

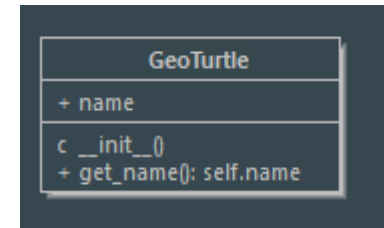
Kleinere hilfreiche Features

- Absatzmarken



```
class GeoTurtle(Turtle):  
    ...  
    def __init__(self):  
        ...  
        self.name = "Testturtle"  
    ...  
    def get_name(self):  
        ...  
        return self.name
```

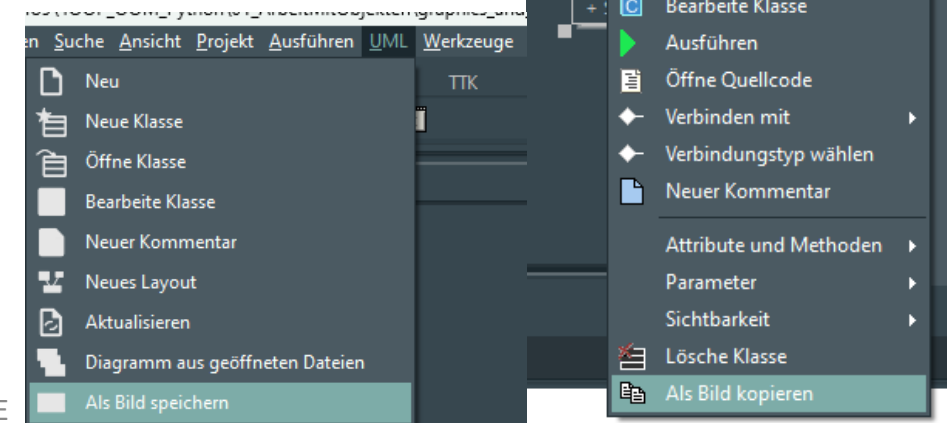
- Klassenkarte aus Editor heraus erzeugen



- Diagramm lässt sich direkt für AB o.Ä. kopieren

- Codefaltung mit 

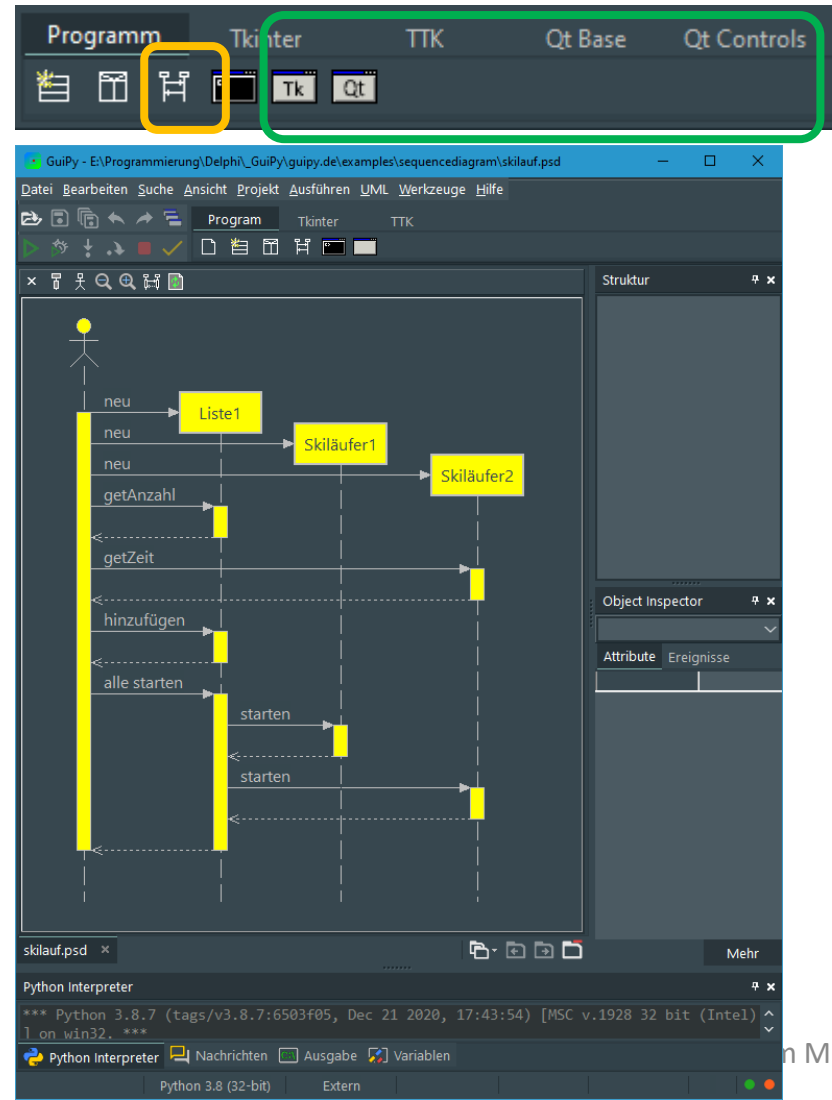
```
class LasercutterTurtle(Turtle):  
    """  
    Lasercutter-Turtle mit mehr Methoden  
    """  
    def SechseckZeichnen(self, kantenlänge = 50):  
        """  
        Zeichnet ein reguläres Sechseck mit angegebener  
        -- Parameter kantenlänge Kantenlänge des ge  
        """  
        self.Gehen(kantenlänge)  
        self.Drehen(60)
```





weitere Funktionen von GuiPy

- GUI-Editor
- Sequenzdiagramm Editor





Werkzeugvergleich – Didaktische Möglichkeiten

Kriterien/Werkzeuge	GuiPy	BlueJ	IDLE	Spyder	Thonny
einfache Handhabung	3	2	1	3	2
Sprache - Deutsch	ja	ja	ja	ja	ja
UML -> Code	ja	nein	nein	nein	nein
Codeblöcke generierbar	ja	nein	nein	nein	nein
Strukturübersicht der Klasse	ja	über Erweiterung	nein	nein	ja
Direkteingabe/Interpreter	ja	ja	ja	ja	ja
Konfigurationmöglichkeiten	sehr vielfältig	wenig bis gar nicht	kaum	vielfältig	wenig

Werkzeugvergleich – Editoreigenschaften

Kriterien/Werkzeuge	GuiPy	BlueJ	IDLE	Spyder	Thonny
farbliche Blockhinterlegung	nein	ja	nein	nein	nein
Code Completion	2-	3	6	2-	1-
Auskommentieren mehrerer Zeilen	ja	ja	ja	ja	ja
Highlighting v. Variablen	ja	ja	nein	nein	ja

Werkzeugvergleich – Modellierung

Kriterien/Werkzeuge	GuiPy	BlueJ	IDLE	Spyder	Thonny
Klassendiagramm	ja	ja	nein	nein	nein
Klassenkarte	ja	über Erweiterung	nein	nein	nein
Objektkarte	jein	ja	nein	nein	nein
Objektdiagramm	jein	nein	nein	nein	nein
Struktogramm	ja	nein	nein	nein	nein
Sequenzdiagramm	ja	nein	nein	nein	nein

Werkzeugvergleich – Editoreigenschaften

Kriterien/Werkzeuge	GuiPy	BlueJ	IDLE	Spyder	Thonny
Plattformunabhängigkeit	jein	ja	ja	ja	ja
Portable	ja	ja	? Wird mitgeliefert	nein	nein
Entwickleranzahl	1	viele	viele	begrenzte Anzahl	viele
Community	begrenzt	weltweit	weltweit	weltweit	weltweit