

IT2School

Gemeinsam IT entdecken



Modul A3 – Programmieren II

Objektorientierte Programmierung mit Python

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Inhalt

1	Objektorientierte Programmierung mit Python	3
2	Warum gibt es das Modul?	4
3	Ziele des Moduls.....	4
4	Die Rolle der Unternehmensvertreterin/des Unternehmensvertreters.....	4
5	Inhalte des Moduls.....	5
5.1	Python	5
5.2	Entwicklungsumgebung Geany	5
6	Unterrichtliche Umsetzung.....	6
6.1	Grober Unterrichtsplan.....	7
6.2	Stundenverlaufsskizzen	8
6.2.1	Einstieg in Python	8
6.2.2	Einführung in die objektorientierte Programmierung.....	8
6.2.3	Entwicklung eines Memory-Spiels.....	9
6.2.4	Vertiefung in die objektorientierte Programmierung.....	10
7	Einbettung in verschiedene Fächer und Themen	11
8	Anschlussthemen.....	11
9	Literatur und Links	12
10	Arbeitsmaterialien	12
11	Glossar	13

1 Objektorientierte Programmierung mit Python

Möchte man größere und aufwendigere Programme schreiben, dann ist es zwingend erforderlich, sich auch tiefergehend mit der Programmierung an sich zu beschäftigen.

Die Schülerinnen und Schüler befassen sich im Rahmen dieses Moduls mit der textuellen Programmiersprache Python und steigen dabei mit einfach Beispielen, wie das Nachprogrammieren eines Programms von Scratch nach Python ein. Im Anschluss daran lernen die Schülerinnen und Schüler Grundkonzepte der Programmierung kennen, wie beispielsweise die objektorientierte Programmierung, Vererbung, Datenkapselung und Entwurfsmuster. Alle Inhalte werden praktisch angewendet, so haben Schülerinnen und Schüler die Möglichkeit eigene Spiele mit einer grafischen Oberfläche zu programmieren



Lernfeld/Cluster:	Daten erforschen	
Zielgruppe/Klassenstufe:		4. bis 5. Klasse
		6. bis 7. Klasse
	X	8. bis 10. Klasse
	X	11. bis 12. Klasse
Geschätzter Zeitaufwand:	8 – 10 Einzelstunden	
Lernziele:	<ul style="list-style-type: none">• Kennenlernen von Python als Programmiersprache.• Überführung einfacher Scratchprogramme in Python.• Prinzip der objektorientierten Programmierung am Beispiel von Python kennenlernen.• Datenkapselung, Vererbung und Entwurfsmuster (speziell MVC) im Zusammenhang mit der objektorientierten Programmierung kennenlernen.• Entwicklung und Ausbau von Programme in Python unter Verwendung der objektorientierten Programmierung.• Gestaltung von grafischen Oberflächen in Python mit der Bibliothek <i>Tkinter</i>.	
Vorkenntnisse der Schülerinnen und Schüler:	Erforderlich: <ul style="list-style-type: none">• Grunderfahrung in der Programmierung mit Scratch	
Vorkenntnisse der/des Lehrenden:	Erforderlich: <ul style="list-style-type: none">• Grunderfahrung in der Programmierung in Scratch Empfohlen: <ul style="list-style-type: none">• Weitere Programmiererfahrung mit anderen Programmiersprachen	

Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters	Keine
Sonstige Voraussetzungen:	Erforderlich: <ul style="list-style-type: none"> • Ausreichende Anzahl an Computern • Python und Geany sollten vorinstalliert sein

2 Warum gibt es das Modul?

Dieses Modul baut auf das Modul *B5 – Programmieren* auf. Die Erfahrungen mit der Programmieroberfläche Scratch werden in diesem Modul aufgegriffen und in eine textuelle Programmiersprache übersetzt. Die Schülerinnen und Schüler erhalten damit einen Einblick in die erweiterte Programmierung und deren komplexe Entwicklung.

Der Schwerpunkt dieses Moduls liegt auf der objektorientierten Programmierung, die ein wichtiger Bestandteil der zeitgemäßen und praxisnahen Anwendungsentwicklung ist. Darüber hinaus wird der Gebrauch von Entwurfsmustern aufgezeigt. Dies ermöglicht es für wiederkehrende Probleme und Anforderungen Lösungsschemata zu verwenden.

Zielsetzung dieses Moduls ist es, die Schülerinnen und Schüler für die Programmierung von komplexeren Anwendungen zu motivieren und ihnen einen praxisnahen Einblick zu gewähren. Durch die Umsetzung von eigenen Projekten können die Schülerinnen und Schüler kreative Ideen umsetzen und die Möglichkeit zur Mitgestaltung in der digitalen Gesellschaft erfahren.

3 Ziele des Moduls

- Kennenlernen von Python als Programmiersprache.
- Überführung einfacher Scratchprogramme in Python.
- Prinzip der objektorientierten Programmierung am Beispiel von Python kennenlernen.
- Datenkapselung, Vererbung und Entwurfsmuster (speziell MVC) im Zusammenhang mit der objektorientierten Programmierung kennenlernen.
- Entwicklung und Ausbau von Programme in Python unter Verwendung der objektorientierten Programmierung.
- Gestaltung von grafischen Oberflächen in Python mit der Bibliothek *Tkinter*..

4 Die Rolle der Unternehmensvertreterin/des Unternehmensvertreters

Im *Modul A3 – Programmieren II* hat die Unternehmensvertreterin/der Unternehmensvertreter mehrere Möglichkeiten aktiv mitzuwirken. Hier einige Anregungen:

- Unterstützung der Lehrkraft, z. B. bei der Einführung in die Programmierung.

- Unterstützung der Schülerinnen und Schüler bei der Umsetzung ihrer eigenen Projekte.
- Sie oder er kann zur Abschlusspräsentation der Projektergebnisse als Special-Guest eingeladen werden.

5 Inhalte des Moduls

5.1 Python

Bei Python handelt es sich, ähnlich wie bei Scratch aus B5 Programmieren, um eine Programmiersprache. Anders als Scratch handelt es sich bei Python jedoch um eine textuelle Programmiersprache, die universell einsetzbar ist und auf verschiedenen Systemen Anwendung findet. Dies bedeutet, dass die Anweisungen, die vom Programm ausgeführt werden, in Textform (dem sogenannten Quellcode bzw. Sourcecode) verfasst wird. Anschließend kann dieser Quellcode auf jedem System, das über die Programmiersprache (bzw. den Interpreten) Python verfügt, ausgeführt werden.

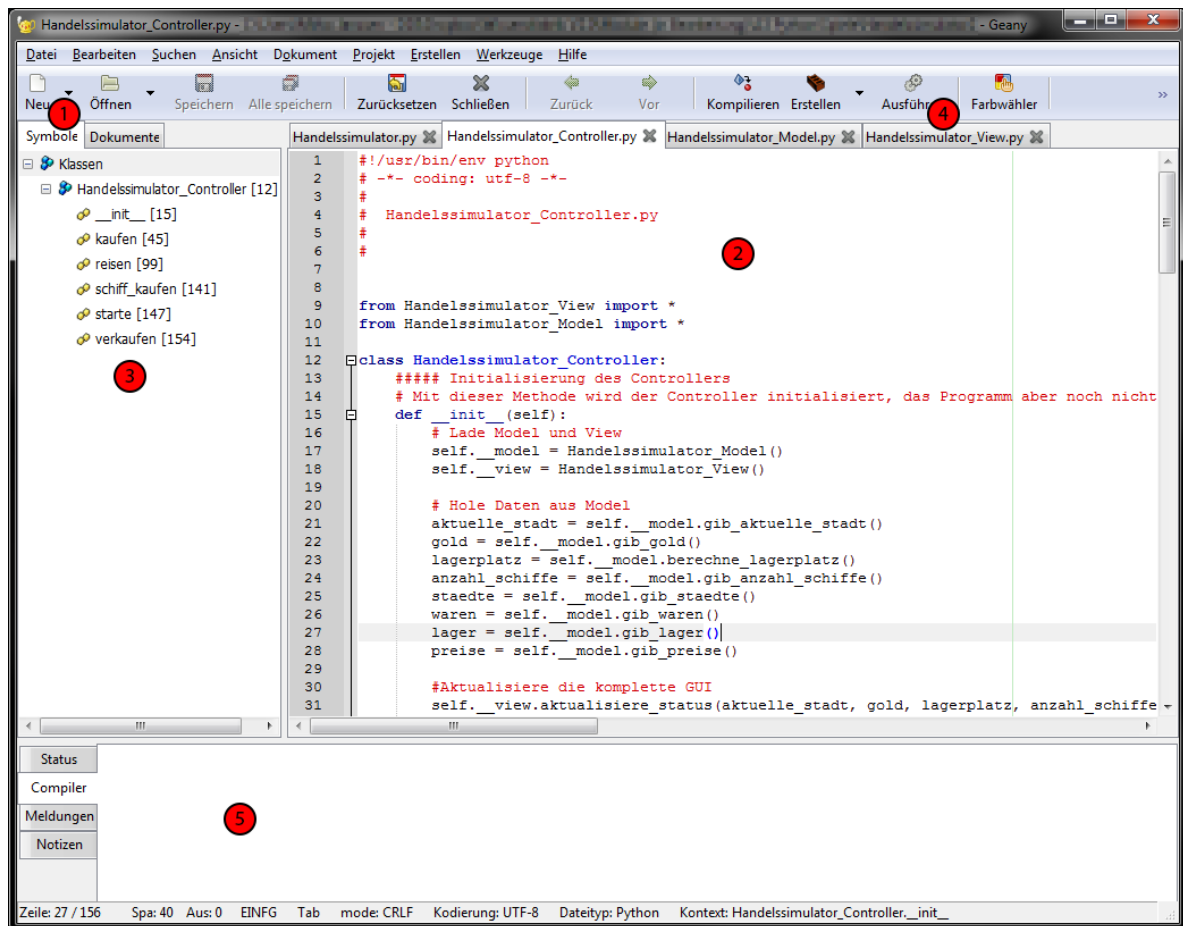
Für die Entwicklung von Programmen existierten viele verschiedene Umgebungen. In diesem Modul beschränken wir uns auf die Entwicklungsumgebung Geany. Jedoch ist es auch möglich, das Programm mit dem ganz gewöhnlichen Editor (nicht zu verwechseln mit Word oder Wordpad) zu schreiben und dem Standardterminal unter Windows, Linux etc. auszuführen.

Python liegt in zwei Versionen vor und kann von der offiziellen Webseite <http://python.org/download/> ganz einfach als Setup runtergeladen und installiert werden. Wir verwenden die Version 2.7 in diesem Modul.

Im Gegensatz zu anderen Programmiersprachen werden in Python Befehle nicht durch ein Semikolon beendet und Schleifen oder Bedingungen durch geschweifte Klammern eingegrenzt. Im letzten Fall werden stattdessen die entsprechenden Anweisungen eingerückt. Dies soll dafür sorgen, dass der Quellcode im Vergleich zu anderen Sprachen möglichst lesbar und damit anfängerfreundlich ist.

5.2 Entwicklungsumgebung Geany

Geany ist eine Entwicklungsumgebung für verschiedene Programmiersprachen (darunter auch Python) und kann unter Linux, OS X und Windows verwendet werden. Die Entwicklungsumgebung besitzt hilfreiche Funktionen, wie zum Beispiel Übersichten von Methoden oder Autovervollständigung. Geany lässt sich dabei ganz einfach wie übliche Entwicklungsumgebungen benutzen. Es folgen eine Grafik der Entwicklungsumgebung, inklusiver der Kennzeichnung einiger wichtiger Elemente sowie deren Erläuterung:



1. Hier kann eine neue Datei angelegt werden. Wichtig zu erwähnen ist hier, dass verschiedene Formate (.py für Python) als Vorlagen ausgewählt werden können.
2. In diesem Bereich befindet sich der Quellcode des Programms.
3. Hier werden die Klassen und deren Methoden aus der aktuellen Datei aufgelistet.
4. Mit diesem Button kann das Programm ausgeführt werden.
5. In diesem Bereich werden alle Benachrichtigungen in Bezug auf das Kompilieren bzw. Ausführen des Programmes angezeigt. Fehlermeldungen werden grundsätzlich im Terminal des ausgeführten Programmes angezeigt.

6 Unterrichtliche Umsetzung

In diesem Modul steht die selbstständige Arbeit der Schülerinnen und Schüler im Vordergrund. Zu Beginn werden die Schülerinnen und Schüler angeleitet, neue Konzepte zu lernen. Danach können die Schülerinnen und Schülern sich eigenverantwortlich in die Thematik vertiefen und auch Arbeitsaufträge umsetzen.

Als Lehrkraft sind sie daher zum großen Teil als Lernbegleitung im Einsatz. Dies bedeutet, dass sie Materialien und Aufgabenstellungen vorgeben, jedoch bei der Problemlösung nur wenig eingreifen bzw. den Weg zur Lösung nicht direkt vorgeben.

Nach Abschluss des Moduls verfügen die Schülerinnen und Schüler über ein erweitertes Verständnis bzgl. der Programmierung (im Vergleich zu dem Modul B5). Die Schülerinnen und Schüler sind nun in der Lage, eigene Projekte und Ideen zu entwickeln und umzusetzen.

6.1 Grober Unterrichtsplan

Unterrichtsszenarien	Kurze Zusammenfassung
Einstieg	Python sowie Geany werden vorgestellt und die Schülerinnen und Schüler überführen ein Programm von Scratch nach Python.
Erarbeitung	Die Schülerinnen und Schüler machen erste Erfahrungen mit dem Konzept der objektorientierten Programmierung und der Entwicklung von grafischen Oberflächen.
Vertiefung	Die Schülerinnen und Schüler programmieren ohne direkte Anleitung ein Memory mit grafischer Oberfläche.
Erarbeitung	Das Konzept der objektorientierten Programmierung wird vertieft und Themen wie Datenkapselung, Vererbung und Entwurfsmuster werden behandelt.
Vertiefung	Die Schülerinnen und Schüler arbeiten sich in ein vorgegebenes unfertiges Spiel (Handelssimulator) ein und stellen dieses fertig.

6.2 Stundenverlaufsskizzen

Abkürzungen/Legende

AB = Arbeitsblatt/Arbeitsblätter; L = Lehrkraft; MuM = Mitschülerinnen und Mitschüler; SuS = Schülerinnen und Schüler;

UV = Unternehmensvertreterin/Unternehmensvertreter

6.2.1 Einstieg in Python

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
5 Min.	Einstieg	Plenum	Begrüßung; Vorstellung von Python und Geany	
25 Min.	Erarbeitung	Einzel-/Partnerarbeit	Die SuS befassen sich mit A3.1 beschäftigen. L steht bei Problemen mit der Entwicklungsumgebung zur Verfügung. Wichtig ist, dass die L nicht direkt Lösungen vorgibt, da die Anleitung zum einen das Machen eines bestimmten Fehlers vorsieht und zum anderen sehr detailliert ist. Zum Abschluss kann der Lehrer auch die Befehlsübersicht A3.8 austeilen.	A3.1, Computer, ggf. A3.8
15 Min.	Abschluss	Plenum	Einige Programme können von den SuS präsentiert und Aufgabe 4 kann besprochen werden. Zum Abschluss kann die L noch einen Überblick über die kommenden Stunden geben.	Computer, Beamer

6.2.2 Einführung in die objektorientierte Programmierung

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
5 Min.	Einstieg	Plenum	Begrüßung; Ablauf der Unterrichtsstunde klären	

15 Min.	Erarbeitung	Lehrervortrag oder Einzel-/Partnerarbeit	Einstieg in die objektorientierter Programmierung; Möglich durch Lehrervortrag, oder durch Austeilen und selbstständiges Arbeiten mittels A3.2. Hier sollte deutlich werden, dass die objektorientierte Programmierung wichtig für die kommenden Stunden ist: Bei der Programmierung einer grafischen Oberfläche wird ausschließlich mit Klassen und Objekten gearbeitet!	A3.2
20 Min.	Vertiefung	Einzel-/Partnerarbeit	SuS arbeiten sich in die Grundlage der grafischen Oberflächen mittels des Tutorials A3.3 ein.	A3.3, Computer
5 Min.	Abschluss	Plenum	Im Plenum sollte der Inhalt der Unterrichtsstunde kurz reflektiert und besprochen werden. Ein Ausblick auf die kommende Stunde kann ebenfalls gegeben werden.	

6.2.3 Entwicklung eines Memory-Spiels

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
5 Min.	Einstieg	Plenum	Begrüßung; Besprechung des Ablaufs dieser und nächsten Unterrichtsstunde	
70 Min. ¹	Erarbeitung	Einzel-/Partnerarbeit	Die SuS entwickeln selbstständig ein eigenes Memory-Spiel. Als Grundlage erhalten sie das AB A3.4 und das ZIP-Archiv A3.5. Das AB A3.4 bietet ein paar Hilfestellungen für mögliche Probleme, ebenso wie die L als Hilfestellung bereitstehen sollte.	A3.4, A3.5, Computer
15 Min.	Abschluss	Plenum	Zum Abschluss können alle oder einige Programme im Plenum präsentiert werden. Es ist auch möglich, die verschiedenen Programm zum Austausch bereitzustellen.	

¹ Wir gehen hier handlicherweise von einer Doppelstunde aus, jedoch kann der Inhalt auch auf zwei Einzelstunden verteilt werden.

6.2.4 Vertiefung in die objektorientierte Programmierung

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
5 Min.	Einstieg	Plenum	Begrüßung; Ablauf der Unterrichtsstunde klären	
20 Min.	Erarbeitung	Lehrervortrag oder Einzel-/Partnerarbeit	Einstieg in den zweiten Teil der objektorientierter Programmierung; Möglich durch Lehrervortrag, oder durch Austeilen und selbstständiges Arbeiten mittels A3.6. Hier sollte deutlich werden, dass die objektorientierte Programmierung wichtig für die kommenden Stunden ist: Bei der Programmierung einer grafischen Oberfläche wird ausschließlich mit Klassen und Objekten gearbeitet!	A3.6
95 Min. ²	Erarbeitung	Einzel-/Partnerarbeit	Die SuS arbeiten sich in den Handelsimulator ein und erweitern diesen.	A3.7, Computer
15 Min.	Abschluss	Plenum	Zum Abschluss können alle oder einige Programme im Plenum präsentiert werden. Es ist auch möglich, die verschiedenen Programme zum Austausch bereitzustellen.	

² Bei einer starken Klasse könnten auch nur zwei Unterrichtsstunden reichen, jedoch sind drei zu empfehlen.



7 Einbettung in verschiedene Fächer und Themen

Als Einbettung in ein spezielles Unterrichtsfach bietet sich in erster Linie die Informatik an. Dies liegt vor allem in der Tatsache, dass es sich bei diesem Modul um eine Vertiefung der Programmierung handelt. Da gerade die Informatik nicht in allen Bundesländern fester Bestandteil der Schulbildung ist und einige Schulen kein Unterrichtsfach in Richtung der Informatik anbieten, würde es sich als Alternative anbieten, dieses Modul fächerübergreifend im Rahmen einer Projekt- bzw. Themenwoche einzubinden.

Die folgenden Kompetenzen finden sich entweder in den Bildungsstandards der Kultusministerkonferenz oder in den einzelnen Rahmenlehrplänen der Länder wieder:

Informatik

Die Schülerinnen und Schüler ...

- erstellen Produkte unter Anwendung fortgeschrittener Techniken von Standardsoftware; falls es das Produkt erfordert, arbeiten sie sich in geringem Umfang in Spezialsoftware ein.
- erwerben beim Bearbeiten von Softwareprojekten in angemessenem Umfang Kenntnisse über Analyse- und Modellierungsverfahren sowie Projektmanagement.
- verstehen Programmabläufe und die Arbeitsweise von Schnittstellen.
- können erdachte Systeme in technische Systeme übertragen.
- kennen sich in Entwicklungsumgebungen/Programmierungsumgebungen aus.
- eine Spielumgebung entsprechend der Spielidee gestalten.
- eine Spielvariante mit informatischen Werkzeugen entwerfen und implementieren.

8 Anschluss Themen

Als Anschluss Themen im Zusammenhang mit IT2School bietet sich folgendes Modul an:

Beispiel: Ein Einstieg in eine weitere textuelle Programmiersprache

Im Erweiterungsmodul E3 Robotik haben die Schülerinnen und Schüler die Möglichkeit einen eigenen BB8 (ein Roboter aus dem Star Wars Universum) zu bauen. Bei diesem Bauvorhaben kann auf den Arduino zurückgegriffen werden. Dies bietet die Möglichkeit sich nochmals einer anderen textuellen Programmiersprache zu widmen. Ein wichtiger Unterschied im Vergleich zu der bisherigen Programmierung ist dann, dass mit der dort verwendeten Programmiersprache unter anderem Motoren angesteuert werden können.









Beispiel: Verschiedene Vertiefungsmöglichkeiten




Verfügen die Schülerinnen und Schüler ersteinmal über ein gewisses Potenzial in der Programmierung, so kann hierauf hinsichtlich verschiedener Vertiefungsmöglichkeiten aufgebaut werden. So wäre es beispielsweise möglich sich intensiver mit dem Entwicklungsprozess bzw. der Planung und Konzeption von Anwendungen zu beschäftigen. Schülerinnen und Schüler könnten sich in dem Fall in dem Bereich von UML-Diagrammen oder Ähnlichem vertiefen. Eine weitere Vertiefungsmöglichkeit stellt auch das Thema Datenbanken dar. Hier könnten die Schülerinnen und Schüler eigene Datenbanken für gewisse Aufgabenstellungen entwerfen und anschließend in ihre Programme integrieren.

9 Literatur und Links




- Glöde, Martina & Reit, Birgit (Hg.) (2015): **Programmieren supereasy. Einfacher Einstieg in Scratch und Python**. München: Dorling Kindersley.
- Onlie-Kurs bzw. -Tutorial zu den Grundlagen von Python – online: <http://www.python-kurs.eu/kurs.php>
- **Online-Kurs bzw. -Tutorial zu den Grundlagen der Gestaltung von grafischen Oberflächen mit Tkinter in Python** – online: <http://www.tkdcs.com/tutorial/index.html>
- **Lernplattform für Python** sowie die Auswertung von Daten mittels Python – online: <https://www.datacamp.com>

10 Arbeitsmaterialien

Nr.	Titel	Beschreibung
 A3.1	Einstieg	Bietet einen Einstieg in Python, wobei ein Programm/Spiel von Scratch in Python überführt wird.
 A3.2	Objektorientierte Programmierung – Teil 1	Einstieg in die objektorientierte Programmierung. Erklärt, was es ist und gibt Beispiele zur Verwendung von Klassen und Objekten.
 A3.3	Einstieg grafische Oberflächen	Vorstellung der Bibliothek Tkinter zur Gestaltung von grafischen Oberflächen. Enthält auch ein Beispiel Programm als Tutorial.
 A3.4	Memory	Aufgabenstellung zur Gestaltung eines eigenen Memorys. Beinhaltet auch Problemstellungen inklusive deren Lösungen.
 A3.5	Memory – Vorlage	Vorlage für Schülerinnen und Schüler zur Aufgabenstellung in A3.4.
 A3.6	Objektorientierte Programmierung – Teil 2	Der zweite Teil der objektorientierten Programmierung wird vorgestellt. Inhaltlich umfasst dieses Material die Vererbung, Datenkapselung und das Entwurfsmuster MVC.

 A3.7	Handelssimulator	Aufgabenstellung zur Einarbeitung in eine Rohvariante des Handelssimulator sowie der zusätzlichen Aufgabenstellung zur Erweiterung dieses Programms/Spiels.
 A3.8	Befehlsübersicht	Eine Gegenüberstellung einiger Bausteine aus Scratch zu den Befehlen aus Python.
 A3.9	Installationsanleitung	Kurze Einführung in Python und Geany sowie eine Installationsanleitung.

Legende

-  Material für Schülerinnen und Schüler
-  Material für Lehrkräfte sowie Unternehmensvertreterinnen und Unternehmensvertreter
-  Zusatzmaterial

11 Glossar

Begriff	Erläuterung
Datenkapselung	Bezeichnet die Zugriffsbegrenzung von Attribute (und damit auch werden Daten/werten). Bei der Definition einer Klasse können deren Attribute als öffentlich, privat oder geschützt gekennzeichnet werden.
Entwurfsmuster	Als Entwurfsmuster werden Lösungsschablonen zu wiederkehrenden Entwicklungsproblemen bezeichnet. Dies kann zum Beispiel das Problem bzw. der Wunsch sein, dann bei der Entwicklung einer Anwendung die Logik, die Datenquellen sowie die grafischen Oberflächen im Quellcode von einander getrennt werden.
Geany	Eine Entwicklungsumgebung für mehrere Programmiersprachen, so auch für Python.
MVC	Ein Entwurfsmuster, dann bei der Entwicklung einer Anwendung zum Einsatz kommt, wenn die Logik, die Datenquellen sowie die grafischen Oberflächen im Quellcode von einander getrennt werden sollen.
Objektorientierte Programmierung	Ein Paradigma der Programmierung, bei dem Objekte und Klassen als Grundstrukturen der Wirklichkeit in der Konzeption von Anwendungen entworfen und verwendet werden.
Scratch	Eine blockbasierte Programmiersprache, welche sehr gut für junge Schülerinnen und Schüler als Einstieg in die Programmierung geeignet ist.
Vererbung	Ein grundlegenden Konzepte der objektorientierten Programmierung, welche Beziehungen zwischen

	Klassen/Objekten zulässt. Methoden und Attribute können dabei von der Mutterklasse auch die abgeleiteten Klasse vererbt werden.
--	---