



**Fonds für Unterrichts- und Schulentwicklung
(IMST-Fonds)**

S2 „Grundbildung und Standards“

IN DER UNTERSTUFE SPIELERISCH PROGRAMMIEREN LERNEN MIT JAVAKARA UND ROBOT KAROL

ID 556

FI Mag. Günther Schwarz

Mag. Christa Leonhartsberger, BG Ried i.I.

Mag. Thomas Kroupa, Gymnasium Dachsberg

Mag. Florian Hutterer, BG Schärding

Mag. Johann Eizinger, BRG Schloss Wagrain, Vöcklabruck

Mag. Josef Pührer, BG Bad Ischl

Mag. DI Christian Kloimböck, Ramsauergymnasium

Mag. Marisa Herbst, BG Schloss Traunsee

Vöcklabruck, Juli 2007

INHALTSVERZEICHNIS

ABSTRACT	4
1 EINLEITUNG	5
1.1 Ausgangssituation.....	5
1.2 Ziele	5
1.3 Grundbildungskonzept	5
1.4 Überprüfung des Erreichens der Ziele	5
2 PROGRAMMIERUMGEBUNGEN	6
2.1 Robot Karol	6
2.2 Java Kara.....	7
2.3 Logo	9
3 EINSATZ DER LERNPLATTFORM MOODLE	10
4 DURCHFÜHRUNG DES PROJEKTS IN DEN SCHULEN	12
4.1 Gymnasium Dachsberg.....	12
4.2 BRG Schloss Wagrain, Vöcklabruck.....	12
4.3 Ramsauergymnasium	13
4.4 BG Schärding.....	14
4.5 BG Schloss Traunsee	15
4.6 BG Bad Ischl	15
4.7 BG Ried im Innkreis	16
5 BEFRAGUNG DER SCHÜLERINNEN	17
5.1 Ergebnisse	17
5.1.1 Zusammensetzung der Schülergruppen	17
5.1.2 Programmiererfahrung	18
5.1.3 Veränderungen im Problemlöseverhalten in der Zeit des Programmierunterrichts.....	19
5.1.4 Einstellung zum Programmieren und zum Programmierunterricht.....	23
5.1.5 Einschätzung durch die Lehrerinnen und Lehrer	26
6 ZUSAMMENFASSUNG	29
6.1 Programmierunterricht in der Unterstufe	29
6.2 Einsatz von Programmierlernumgebungen	29

6.3	Einfluss auf die Problemlösekompetenz	29
6.4	Einstellung und Arbeitshaltung der Schülerinnen und Schüler.....	30
6.5	Einstellung zum Programmieren und zum Programmierunterricht.....	31
7	SCHLUSSFOLGERUNGEN UND AUSBLICK.....	32
8	LITERATUR.....	33
	ANHANG.....	34
	Der Fragebogen vor dem Programmierunterricht	34
	Der Fragebogen nach dem Programmierunterricht	41

ABSTRACT

In diesem Projekt ging es um den Programmierunterricht in der Unterstufe. Es wurden einfache Programmierumgebungen (Robot Karol, JavaKara und Logo) im Rahmen von Unverbindlichen Übungen oder Kurzkursen eingesetzt. Beobachtet wurde die Eignung dieser Programmierumgebungen für den Unterricht, die Leistung und das Interesse der Schülerinnen und Schüler und die Auswirkung des Unterrichts auf die Einstellung zum Problemlösen und auf die Problemlösekompetenz.

Das Ergebnis kurz zusammengefasst: Die Programmierumgebungen eignen sich für den Einstieg ins Programmieren. Robot Karol kann zum Einstieg ins Programmieren in 4 bis 6 Doppelstunden sinnvoll genutzt werden. Die Schülerinnen und Schüler lösen damit nur einfachere Aufgaben, komplexere werden nicht in Angriff genommen. Bemängelt wird das Fehlen von Variablen. JavaKara wird in dieser Alterstufe am besten als Automat programmiert und spricht besonders Mädchen an. Java ist wegen der komplizierten Syntax (Klammern) zu schwierig. Logo eignet sich für diese Alterstufe sehr gut. Der Schwierigkeitsgrad der Aufgaben kann allmählich gesteigert werden, die erzeugten Grafiken gefallen den Schüler/innen und Rekursion wird auch für diese Altersstufe erfahrbar.

Das Interesse der Schüler/innen am Programmierunterricht war sehr groß, das Programmieren wurde mit diesen Programmen als leicht empfunden und ein Fortsetzungskurs von den meisten gewünscht. Ein Einfluss des Programmierunterrichtes auf das Problemlöseverhalten und die Problemlösekompetenz konnte aus der Untersuchung nicht abgeleitet werden.

Empfehlung: Es sollte in allen Schulen ein Programmierunterricht mit derartigen Programmierumgebungen in der 3. und/oder 4. Klasse angeboten werden.

Schulstufe: 5.-9.

Fächer: Informatik

Kontaktperson: FI Mag. Günther Schwarz

Kontaktadresse: Landesschulrat für OÖ, Sonnensteinstr. 20, 4040 Linz

1 EINLEITUNG

1.1 Ausgangssituation

In fast allen AHS in OÖ werden in der Unterstufe in Informatik nur Unverbindliche Übungen im Bereich Standardsoftware angeboten. Bei einer Befragung im Rahmen eines MNI – Projekts haben viele SchülerInnen angegeben, dass sie gerne programmieren lernen möchten. Die Erhebung im Rahmen dieses Projekts hat ebenfalls ergeben, dass Schülerinnen und Schüler programmieren lernen möchten. Meine Erfahrung aus dem IT-Unterricht in ersten Klassen ist, dass es in jeder Klasse einige Schüler/innen gibt, die sich gerne mit Lernprogrammen wie Robot Karol oder Java-Kara beschäftigen.

Die Anmeldezahlen für das Wahlpflichtfach Informatik gehen zurück. Schülerinnen und Schüler tun sich schwer, algorithmisch Denken zu lernen. Ein spielerischer Einstieg ins Programmieren in der Unterstufe könnte sich positiv auswirken.

1.2 Ziele

Ziel des Vorhabens war es zu untersuchen, in welcher Alterstufe Schüler/innen algorithmisch denken lernen können und wie sich das Programmieren auf das Problemlöseverhalten der Schüler/innen auswirkt.

Dazu wurde in den teilnehmenden Schulen eine unverbindliche Übung, ein Freigegegenstand oder ein Kurs für Schüler/innen der Unterstufe zum Erlernen des Programmierens angeboten. Es wurden spezielle Programmierumgebungen wie Robot Karol, Java Kara und Logo eingesetzt (Beschreibung im Kapitel 2).

1.3 Grundbildungskonzept

Algorithmen sind ein zentrales und fundamentales Thema der Informatik. Mit Algorithmen Probleme lösen zu können, trägt zur Alltagsbewältigung bei und dient dem Wissenschaftsverständnis. Da Programmieren ein wesentlicher Bestandteil des Informatikstudiums ist, dient dieses Projekt auch der beruflichen Orientierung und Studierfähigkeit.

1.4 Überprüfung des Erreichens der Ziele

Die Schülerinnen und Schüler wurden am Beginn des Projekts und am Ende des Projekts durch einen Fragebogen über ihre Einstellung zum Problemlösen und im speziellen zum Programmieren befragt. Sie mussten außerdem Probleme (Denksportaufgaben) lösen. Dabei wurde ihr Problemlöseverhalten beobachtet.

2 PROGRAMMIERUMGEBUNGEN

Zum Erlernen des Programmierens wurden spezielle Programmierumgebungen und Programmiersprachen eingesetzt, die hier beschrieben werden.

2.1 Robot Karol

Robot Karol ist eine Programmierumgebung mit einer Programmiersprache, die für Schülerinnen und Schüler zum Erlernen des Programmierens und zur Einführung in die Algorithmik gedacht ist.

Robot Karol folgt der Tradition der "Mini-Languages". Dies sind Programmiersprachen, die bewusst über einen kleinen, übersichtlichen Sprachumfang verfügen, um den Einstieg in die Algorithmik zu erleichtern. Dem Programm Robot Karol liegt die Idee von "Karel, the Robot" zugrunde, die zum ersten Mal von Richard E. Pattis in "Karel the Robot: A Gentle Introduction to the Art of Programming" veröffentlicht wurde. Die Idee ist, einen Roboter zu programmieren, der in einer "Bildschirmwelt" lebt. Wenn Karol-Programme ablaufen, sehen die Schülerinnen und Schüler an der Reaktion des Roboters sofort, was sie programmiert haben und ob ihr Programm die Aufgabenstellung erfüllt.

„Robot Karol deutsch“ bietet eine Programmierumgebung mit:

- einem Editor für die Sprache Karol, der Syntaxhervorhebung und Schlüsselwort-Ergänzungen erlaubt
- einer Unterstützung bei der Eingabe des Programmtextes
- einer Darstellung des Programmcodes als Struktogramm und als Code-Übersicht
- einer Syntaxüberprüfung mit Fehlerhinweisen
- einem Programmablauf schnell, verzögert oder mit Einzelschritt
- einer grafischen Darstellung einer 3D-Welt, die den Roboter Karol als Figur im Raum zeigt und diese je nach Anweisungen bewegt
- einer grafischen Darstellung einer 2D-Welt

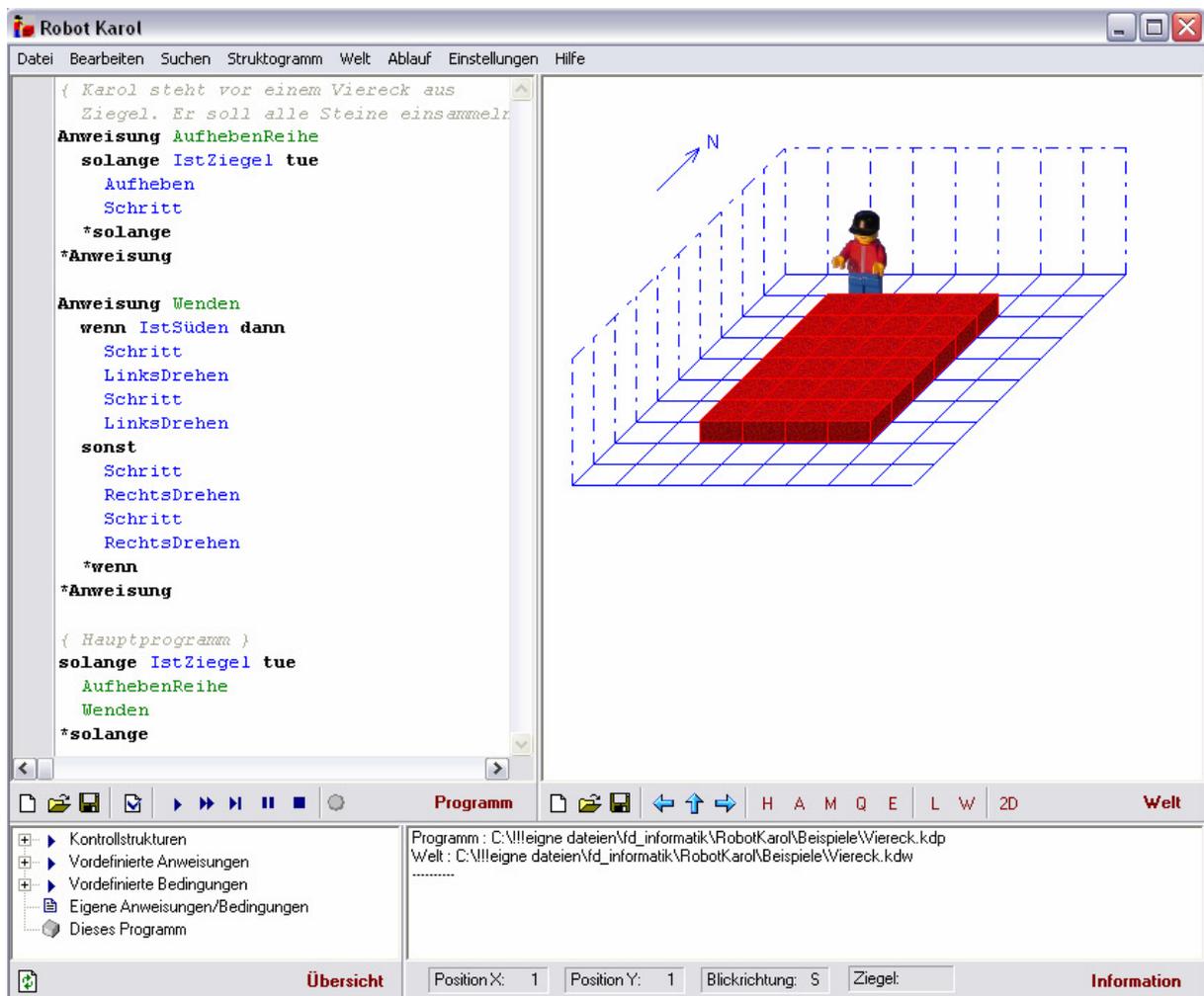
Folgender Screenshot zeigt die Funktionalität der Programmierumgebung. Im linken Fenster wird das Programm geschrieben. Die Befehle können auch durch Betätigung der rechten Maustaste über ein Kontextmenu eingefügt werden (erspart geraden den jungen SchülerInnen die Tipparbeit).

Im rechten Fenster sind die Auswirkungen des Programms zu beobachten. Der Roboter bewegt sich nach den Anweisungen im Programm und legt Steine bzw. hebt diese auf.

Die Schaltflächen unterhalb der Fenster ermöglichen die Steuerung des Programmablaufes und die Steuerung des Roboters von Hand aus. Sie dienen auch zum Löschen der Welt oder zum Wiederherstellen des Anfangszustandes.

Das Programm kann auch in einem automatisch generierten Struktogramm angesehen werden.

Weitere Informationen und Downloadmöglichkeit unter:
<http://www.schule.bayern.de/karol/>



2.2 Java Kara

Kara ist ein Marienkäfer, der in einer einfachen Welt lebt. Er kann programmiert werden und so diverse Aufgaben erledigen. Er kann beispielsweise Kleeblätter sammeln oder hinlegen und Pilze verschieben. Karas Programme sind endliche Automaten und werden in einer grafischen Entwicklungsumgebung erstellt.

Kara vermittelt einen Einstieg in die Grundideen der Programmierung. Zwei Eigenschaften machen den Einstieg mit Kara attraktiv: Endliche Automaten sind einfach zu verstehen, die Einarbeitungszeit ist daher minimal. Zudem arbeitet man bei Kara in einer einfachen Umgebung, ohne mit komplexen Entwicklungsumgebungen konfrontiert zu werden. Auf spielerische Art und Weise machen sich die Schüler/innen mit fundamentalen Ideen der Programmierung vertraut. Kara wird erfolgreich auf verschiedenen Schulstufen (Berufsbildende Schulen, Gymnasien, Fachhochschulen etc.) eingesetzt.

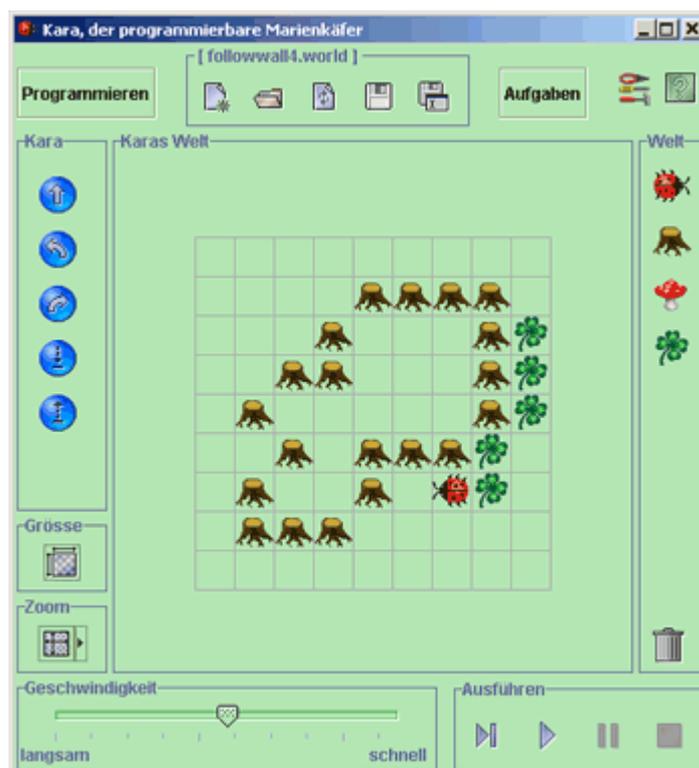
Lernziele beim Einsatz von Kara

» **Grundlagen der Programmierung:** Mit endlichen Automaten kann eine der fundamentalen Ideen von Algorithmen einfach illustriert werden. Ein Programm ist die Spezifikation eines eindeutig bestimmten zeitlichen Ablaufes von Ereignissen – analog zum Regieplan in einem Theater. Die Abläufe müssen auch auf externe Ereignisse reagieren können. Um ein korrektes Systemverhalten zu garantieren, braucht es eine statische Spezifikation, welche die korrekten Abläufe von der noch viel größeren Vielfalt der inkorrekten Abläufe trennt. Kara illustriert diese grundlegende Idee an einfachen Beispielen. Zudem lassen sich auch Konzepte wie Boole'sche Aussagenlogik und Programmverifikation (Pre-Conditions, Invarianten etc.) aufzeigen.

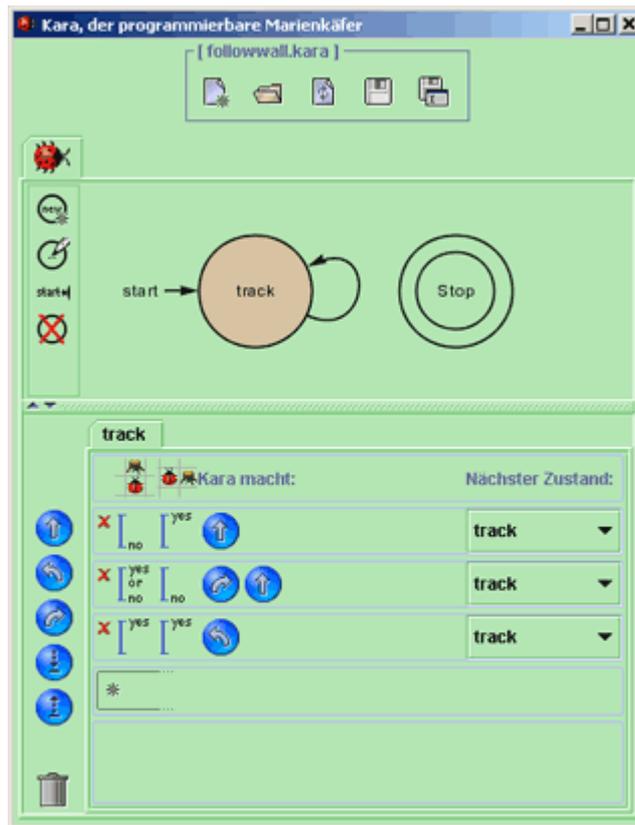
» **Modell der endlichen Automaten.** Endliche Automaten sind eines der einfachsten Berechnungsmodelle der Informatik. Mit ihnen kann das Verhalten vieler aus dem Alltag bekannter Automaten beschrieben werden. Kara veranschaulicht das Konzept endlicher Automaten.

Einige Screenshots:

Im Ausführungsfenster können die Auswirkungen des Programms beobachtet werden. Der Käfer läuft nach den Anweisungen im Programm herum und hebt Kleeblätter auf oder legt sie nieder. Es ist auch eine Steuerung mit den Schaltflächen am linken Rand möglich. Am unteren Rand findet man Schaltflächen zur Steuerung des Programms (Start, Stopp, Geschwindigkeit ändern, ...)



Unterhalb dieses Textes sieht man das Fenster zum Programmieren des Käfers als Automat. Es gibt „Sensoren“ wie „Baum vorne?“, „Baum rechts?“, „Kleeblatt unten?“. Auf das Ergebnis dieser Abfragen (yes – no) wird mit Aktionen reagiert (Schritt vorwärts, links drehen, rechts drehen, Kleeblatt hinlegen, Kleeblatt aufnehmen, ...) und in den nächsten Zustand gewechselt.



2.3 Logo

LOGO ist eine klare und einfache aber dennoch mächtige Programmiersprache. Die Besonderheit von LOGO ist die sogenannte TURTLE-Grafik. Turtlegrafik ist eine verständliche und umfassende Sammlung von Befehlen um eine Turtle (Schildkröte) zu manipulieren, die in der Mitte des Logo -fensters sitzt. Die Turtle selbst kann im Turtle-Fenster Linien zeichnen, wenn sie die entsprechenden Befehle erhält. Die Turtle gibt mit ihren Aktionen dem Programmierer sofort eine Rückmeldung für fast jeden gegebenen Befehl. Dadurch wird es einfacher, programmieren zu lernen: Man gibt der Turtle Befehle, wie man sie einem Zeichen-Roboter, auf dem man mitfahren kann, geben würde und sieht sofort die Auswirkung.

Einige Beispiele:

FD 100	FORWARD 100	bewegt die Turtle um 100 Bildpunkte vorwärts
RT 90	RIGHT 90	dreht die Turtle um 90° nach rechts
LT 90	LEFT 60	dreht die Turtle um 60° nach links
REPEAT 4 [FD 100 RT 90]		REPEAT 4 wiederholt den Befehl in der Klammer 4 Mal, nämlich 100 Bildpunkte vorwärts gehen und dann um 90° nach rechts drehen. Dadurch entsteht ein Rechteck

TO QUADRAT :S

REPEAT 4 [FD :S RT 90]

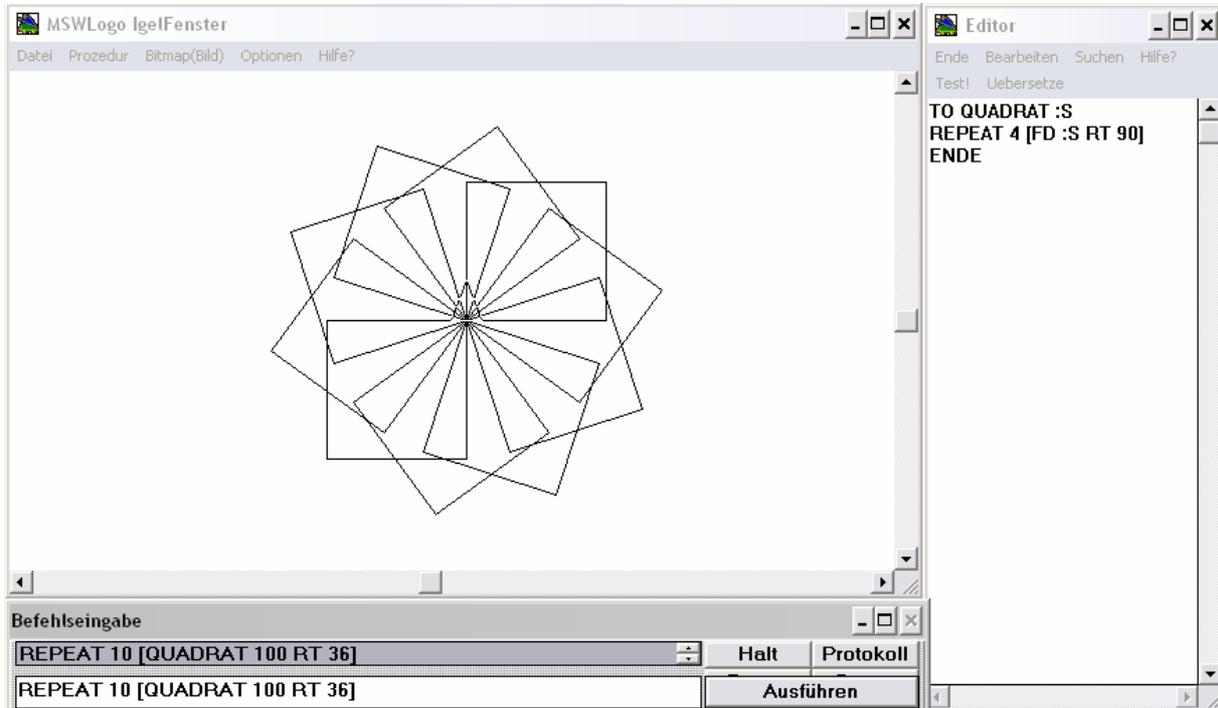
END

Erzeugt einen neuen LOGO Befehl (Prozedur) QUADRAT mit Parameter :S (Ein Doppelpunkt vor einem Buchstaben deklariert eine Variable,

es können damit Quadrate mit variabler Seitenlänge gezeichnet werden)

```
REPEAT 10 [QUADRAT 100 RT 36]
```

Erzeugt untenstehendes Muster



Im BG Ried und im Ramsauergymnasium kam noch der GAMEMAKER zum Einsatz. Im BRG Schloss Wagrain und im Ramsauergymnasium wurden auch Roboter (Lego Mindstorms) programmiert und im BRG Schloss Traunsee programmierten die Schüler/innen zusätzlich mit Visual Basic.

3 EINSATZ DER LERNPLATTFORM MOODLE

Die Lernplattform Moodle wurde zur Kommunikation und zum Austausch von Materialien verwendet. Im Ramsauergymnasium und im Gymnasium Ried diente sie auch zur Unterrichtsorganisation. Dort wurden für die Schülerinnen und Schüler die Informationen und Aufgaben zur Verfügung gestellt und die SchülerInnen konnten die Lösungen abgeben.

Am Beginn des Projekts wurden hier Denksportaufgaben für den Anfangstest und den Endtest gesammelt. Es standen Links zu den Programmen JavaKara und Robot Karol zur Verfügung.

Anleitungen, Kurzfilme und Aufgaben wurden von DI Kloimböck für die Schülerinnen und Schüler erstellt und hier zum Download angeboten.

Screenshot – Moodle – Ramsauergymnasium

Kurs: NMI-Programmieren - Mozilla Firefox

http://www0.bg-rams.ac.at/moodleus/course/view.php?id=12

Sie sind angemeldet als Schwarz Guenther (LogOut)

h50607 >> NMIPR

Personen: Teilnehmer/innen

Aktivitäten: Arbeitsmaterialien, Aufgaben, Feedbacks, Foren

Forensuche: Erweiterte Suche

Administration: Jetzt bearbeiten, Einstellungen, Profil bearbeiten, Lehrer, Schüler, Gruppen, Sicherung, Wiederherstellen, Import, Zurücksetzen, Berichte, Fragen, Bewertungsskalen, Punkte, Dateien, Hilfe

Themen dieses Kurses

Nachrichtenforum

1 Denksportaufgaben

- Link 1
- Link 2
- Link 3
- Denksportaufgaben Eizinger
- Denksportaufgaben Hutterer
- Denksportaufgaben Hutterer
- Denksportaufgaben Kloimböck
- Denksportaufgaben Leonhartsberger
- Denksportaufgaben Pührer

2 JAVAKARA

- JavaKara-Versionen (swisseduc.ch)

3 Robot Karol

- Robot Karol
- moodle-Kurs mit Filmen (Luise-Büchner-Schule)

4 Test Anfang

- Test / Fragebogen Anfang
- Fragebogen Beginn

5 Test Ende

6 Game Maker

- auch eine Möglichkeit den "Kleinen" etwas zum Thema Programmieren zu zeigen (oder von ihnen zu lernen)

Aktuelle Zeit: Server Zeit: 18.07.2007 08:19:54, Lokale Zeit: 18.07.2007 08:11:54

Neueste Nachrichten: Ein neues Thema hinzufügen...

Bald aktuell...: Es gibt keine weiteren Termine

Neueste Aktivitäten: Aktivität seit Monday, 16 July 2007, 08:19

Zur Zeit online: (In den letzten 5 Minuten)

Mag. Christa Leonhartsberger stellte Übungsaufgaben mit Lösungen in Moodle für den Unterricht zusammen. Die Schülerinnen und Schüler konnten die Aufgaben herunterladen, bearbeiten und mit den Lösungen der Lehrerin vergleichen.

Kurs: Robot Karol -

http://www.edumoodle.at/bgried/course/view.php?id=2

Sie sind an

B(R)G Ried >> robot

Personen: Teilnehmer/innen

Aktivitäten: Arbeitsmaterialien, Aufgaben, Foren, Wikis

Suche in Foren: Erweiterte Suche

Administration: Bewertungen, Profil bearbeiten, Kennwort ändern, Ich will mich aus diesem Kurs LOSCHEN!!! Hinweis: Ihre Kursdaten werden damit gelöscht, Meine Kurse: Robot Karol, Gamemaker, Alle Kurse...

Themen dieses Kurses

Ideen, Fragen, Anregungen ...

1

- Hallo, ich bin Robot Karol. In diesem Kurs lernst du mich zu steuern. Du schreibst kleine Programme und ich mache genau das, was du mir anschaffst! Die Programmiersprache Robot Karol kann kostenlos heruntergeladen werden (Unter Links: Robot Karol - Download).
- Übung 1
 - aufg06.kwd
 - aufg07.kwd
 - Lösungen - Übung 1

2 Die ersten einfachen Schleifen- Anweisungen

- Übung 2
 - Lösung Aufgabe 12
 - Lösung Aufgabe 13
 - Lösung Aufgabe 14
 - Lösung Aufgabe 15
 - Lösung Aufgabe 16
 - Lösung Aufgabe 17
 - Lösung Aufgabe 18
 - Lösung Aufgabe 19

Aktuelle Zeit: Server Zeit: 18.07.2007 08:19:54, Lokale Zeit: 18.07.2007 08:11:54

Neueste Nachrichten: Ein neues Thema hinzufügen...

Bald aktuell...: Es gibt keine weiteren Termine

Neueste Aktivitäten: Aktivität seit Monday, 16 July 2007, 08:19

Zur Zeit online: (In den letzten 5 Minuten)

4 DURCHFÜHRUNG DES PROJEKTS IN DEN SCHULEN

In 7 Schule wurde im Rahmen des Unterrichts (Schulautonomer Gegenstand Informatik, Unverbindliche Übung, Freigegegenstand, Kurzkurs) wurden die Programmierumgebungen eingesetzt.

4.1 Gymnasium Dachsberg

Ursprünglich hat sich Mag. Sonja Wiesinger bereit erklärt, am MNI Projekt mitzuwirken. Aus privaten Gründen konnte sie die Zeit dafür nicht aufbringen. So übernahm Mag. Thomas Kroupa die Aufgabe.

Im Gymnasium Dachsberg gibt es einen schulautonomen Pflichtgegenstand Informatik in der 3. und 4. Klasse. In einem Teil dieses Unterrichts wurde Robot Karol und Java Kara eingesetzt.

Resümee von Mag. Thomas Krupa (E-Mail)

Das Projekt ist bei uns sehr gut gelaufen, und wir haben vor, es auch nächstes Jahr fix in unseren Lehrplan der 4. Klasse einzupflegen, da wir bereits in der 3. und 4. Klasse verpflichtende Informatikstunden haben.

Die Schüler waren durchwegs begeistert und hatten viel Spaß. Aufgefallen sind die sehr unterschiedlichen Lösungsansätze der einzelnen Schüler/innen und bei vielen auch ein Verständnis für die Konzepte (Kontrollstrukturen) der Programmierung im Allgemeinen.

4.2 BRG Schloss Wagrain, Vöcklabruck

Mag. Johan Eizinger hat mit einer ersten Klasse gearbeitet. Er begann mit Robot Karol (5 Doppelstunden) und stieg dann um auf Logo um (5 Doppelstunden) und beschäftigte sich anschließend mit der Roboterprogrammierung (Lego Mindstorms). Es zeigte sich, dass das Abstraktionsvermögen der 10 jährigen noch nicht sehr groß ist. Einfache Aufgaben wurden mit Begeisterung gelöst, bei schwierigeren Beispielen stiegen die meisten Schüler/innen aus.

Der Umstieg auf LOGO hat den Schülerinnen und Schülern sehr gut gefallen. Die grafischen Beispiele haben sie sehr angesprochen. Ein paar Schüler haben auch versucht rekursive Programme zu schreiben.

Resümee von Mag. Johann Eizinger

In der ersten Klasse sind die meisten Schüler zwar begeistert bei der Sache, aber nicht sehr ausdauernd. Sie geben rasch auf bzw. verwenden umständlich schon bekannte Strukturen, statt neue Gedanken umzusetzen (z. B. Befehle mehrmals hintereinander anschreiben anstelle einer Wiederholungsanweisung, sobald es nicht mehr ganz trivial ist).

Bei Robot-Karol stößt man bald an die Grenzen der Programmiersprache. Die Beschränkung der Sprache auf wenige Elemente ist zwar „logisch elegant“ (Vergleich: Mathematik – minimale Axiomensysteme bzw. möglichst wenige Voraussetzungen für die Gültigkeit von Sätzen ...), führt aber umso rascher zu sehr umständlichen (um nicht zu sagen: undurchschaubaren) Lösungsansätzen.

Hauptproblem: die nicht vorhandenen Variablen

Bei Logo (mit Turtle-Grafik) war die Experimentierbereitschaft größer und das Lernen neuer Strukturen erfolgte bereitwilliger. Die Ergebnisse waren auch großteils motivierender.

Robolab (graphische Programmierung des Legoroboters)

Die gleichen Schüler, die z. B. bei Logo vergessen, eine Klammer zu schließen, oder statt einer eckigen eine runde Klammer machen, sind es, die bei Robolab irgendeine Leitung vergessen oder die Symbole in der falschen Reihenfolge verbinden etc.

4.3 Ramsauergymnasium

DI Christian Kloimböck hielt 2 Kurse in 3. Klassen. Er begann mit Robot Karol (3-4 Doppelstunden) und bemerkte, dass das Interesse nachließ. Er stieg um auf JavaKara (Programmierung als Automat) und es gelang ihm, die Freude am Programmieren wieder zu wecken. In einer Gruppe wurde noch der Gammaker eingesetzt, der die Schüler sehr angesprochen hat. Insgesamt wurden 12 Doppelstunden durchgeführt.

Resümee DI Kloimböck

Auf Grund der hohen Anzahl von interessierten Schülern der 7.Schulstufe habe ich an unserer Schule zwei solcher Programmierkurse angeboten. Jener am Mittwoch lief von Ende November bis Ende März. Der zweite startete wegen einer Kollision mit einem anderen Kurs später.

Im 1. Kurs wurden ca. 6 Treffen für Robot Karol, 4 für Kara und 2 für Lego-Mindstorms verwendet.

Im 2.Kurs kürzte ich aus den Erfahrungen vom 1.Kurs die Zeit mit Robot Karol ab und stieg kurz auf Lego-Mindstorms um und wechselte dann zu Gammaker. Dann kam/kommt noch Kara.

Ein paar Gedanken, Beobachtungen, Rückmeldungen von Schülern:

- Generell kamen merkliche Unterschiede beim Lösen der Aufgaben zu Tage. Manche Schüler strukturierten das gestellte Problem und konnten so die Auf-

- gaben bewältigen. Andere hingegen hatten auch mit zu bereits gelösten Aufgaben verwandten Beispielen Probleme bzw. konnten sie alleine nicht lösen.
- Einstieg mit Robot Karol ist meist „gut“ gelungen. Bei fortgeschrittenen Aufgaben waren viele Schüler demotiviert (überfordert?) – vielleicht wäre eine längere „Übungsphase“ hilfreich (Kurs übers ganze Jahr?)
 - Das Lösen der Aufgaben mit Kara schien weniger Probleme zu machen als bei jenen mit Robot Karol. Entweder lag es an der Programmierumgebung oder an den von mir gestellten Aufgaben bei RK.
 - Es ist klar, dass ein Teil des Reizes mit Gamemaker zu arbeiten sicher am „Game“ liegt, wobei jedoch etliche Schüler sich durchaus mit der Programmierung ernsthaft auseinandergesetzt haben. Einen Vorteil bringt Gamemaker auch mit sich. Das Produkt wird gerne von Kollegen getestet.
 - „Ist der Kurs eh nicht so schwer wie die Denksportaufgaben zu Beginn?“
 - Gegen Ende der Doppelstunde nahm die Konzentration bei den meisten Schülern ab.

4.4 BG Schärding

Mag. Florian Hutterer hat eine unverbindliche Übung angeboten, an der Schülerinnen und Schüler von der 1. bis zur 4. Klasse teilgenommen haben. Teilweise wurden die Schüler der 4. Klasse als Tutoren eingesetzt. Er hatte 36 Anmeldungen und führte die Übung in 2 Gruppen durch. In einer Gruppe wurden zwei Drittel der Zeit Robot Karol eingesetzt und dann Logo. In der anderen Gruppe war es umgekehrt.

Für die Schülerinnen und Schüler war der Sprung von einfachen Programmen zu komplexeren schwierig. Logo bietet seiner Meinung nach mehr Möglichkeiten, die Komplexität allmählich zu steigern.

Am Anfang wurden die Denksportaufgaben bearbeitet. Seiner Meinung nach ist das Problemlöseinteresse gestiegen.

Der angebotene Moodle-Kurs wurde nur wenig genutzt und nur wenige haben Hausübungen gemacht.

Es wurden 12 Doppelstunden pro Gruppe gehalten.

Zusammenfassende Kommentare zum MNI-Projekt

„Programmieren in der Unterstufe“

Wir starteten jede Unterrichtseinheit mit einem / mehreren Rätsel; über den Verlauf hinweg kann ich behaupten, dass das „Problem lösen“ als Interesse mit der Bewältigung von Problemen steigt, was als Chance zu betrachten ist.

Je kreativer und offener die Aufgabenstellung war, desto besser und umfangreicher waren die Ergebnisse.

Die grafische Aufmachung einer Programmierumgebung stellt für Schüler in sehr wichtiges Kriterium dar. So ist es beispielsweise nicht einfach, nach R. Karol zu Turt-

le-grafiken zu wechseln, da der Verlust einer bunten Welt und einer Dimension schülerseitig nicht zu unterschätzen ist.

Als Unverbindliche Übung für die gesamte Unterstufe (5. bis 8. Schulstufe) stellt sich das Problem, dass die Homogenität der Lerngruppe in Bezug auf logisches Denken, und Denkmuster nicht mehr vorhanden ist.

Die Auswertung der Beispiele beweist, dass die SchülerInnen mehr probiert und versucht haben. Beim Abschlusstest wurden bei mehr Aufgaben etwas notiert (auch wenn dies nur Ansätze oder Versuche waren).

Auch die Anzahl der total korrekt beantworteten Fragebogenbeispiele ist gestiegen.

Für mich steht weiterhin in Frage, ob es sinnvoll ist, komplexe informationstechnische Konzepte (wie beispielsweise die Rekursion) durch spielerische Konzepte in diesem Alter zu vermitteln, da der Transfer hiermit noch nicht garantiert ist und die Komplexität solcher Mechanismen kein Unterstufenthema sein kann - was aber nicht in Frage stellt, dass eine unverbindliche Übung wie diese das Interesse am Computer steigern kann.

Eine didaktische Reduktion (wie der Verzicht auf Variablen bei Robot Karol) garantiert noch nicht ein besseres Verständnis, sondern beschränkt vielmehr oft die Möglichkeiten.

Aus meiner statistischen Auswertung habe ich herausgelesen:

- ✓ Interesse an mathematischen Aufgaben und deren Lösung steigt mit jedem neuen Beispiel
- ✓ Interesse an Spielen ist gestiegen (vielleicht durch Kommunikation der Schüler?)

4.5 BG Schloss Traunsee

Mag. Marisa Herbst hat in 2 Gruppen mit jeweils 10 Schülerinnen und Schülern einer 4. Klasse gearbeitet (1 Doppelstunden pro Woche). Sie hat mit LOGO begonnen. Die Schülerinnen und Schüler waren mit Begeisterung dabei. Die Ergebnisse waren ihnen wichtiger als das Programmieren.

Java Kara Programmierung in Java war für die Schüler/innen zu schwierig (Syntax, Klammern).

Fortgesetzt wurden mit Visual Basic, das ein halbes Jahr verwendet wurde. Ungefähr die Hälfte der SchülerInnen hat wirklich programmiert.

Es wurden Abschlussprojekte durchgeführt. Beispiele: Quiz, Unterwasserwelt-Fisch, Sokoban, Ballspiel.

4.6 BG Bad Ischl

Mag Josef Pührer arbeitete mit 11 Schülern einer 4. Klasse. Er begann mit Robot Karol und stieg dann um auf Java Kara. Das Interesse der Schüler war sehr groß und

sie haben sehr gut mitgearbeitet. Er hat in Supplierstunden diese Programme eingesetzt und festgestellt, dass in Mädchenklassen Java-Kara besser angekommen ist.

Das Interesse war bis zum Schluss sehr gut. Er hat festgestellt, dass die Hälfte der Schüler jetzt in die HTL wechselt. Wie weit diese Entscheidung vom Programmierunterricht beeinflusst worden ist, wurde nicht untersucht.

Resümee Mag. Josef Pührer:

Die Arbeit mit der Gruppe war für mich sehr erfüllend, im normalen Unterricht erlebt man kaum dieses Interesse und Engagement.

Ich befürworte auf jeden Fall das Programmieren in der Unterstufe, weil es neben anderen positiven Aspekten rechtzeitig die richtigen Signale für die Berufswahl setzen kann.

4.7 BG Ried im Innkreis

Mag. Christa Leonhartsberger hat mit 21 Schülerinnen und Schülern von 27 einer ersten Klasse gearbeitet. 8 Doppelstunden wurde Robot Karol eingesetzt, 4 Doppelstunden der Gamemaker.

Den Schülerinnen und Schülern hat es sehr gut gefallen. Für 10-jährige sind die Beispiele zu schwierig geworden. Der Umstieg auf den Gammaker hat die Schülerinnen und Schüler wieder motiviert.

Es wurde auch das Arbeiten mit der Lernplattform Moodle geübt. Die Kombination mit der Lernplattform hat sich bewährt.

Bemerkenswert ist die Teamaufgabe, bei der jeweils 2 Schüler ein Zimmer mit Robot Karol gebaut haben. Diese wurden dann zu einem Haus zusammengebaut.

Bemerkungen von Mag. Christa Leonhartsberger:

Interessant ist, dass alle Schüler die Frage "Wenn dir die Lösung nicht gleich einfällt, gibst du schnell auf?" verneinen. Bei einfachen linearen Programmieraufgaben war dies auch der Fall, bei schwierigen Aufgaben hat ungefähr die Hälfte doch relativ schnell aufgegeben.

Antworten zu "Warum möchtest du programmieren lernen?":

- Weil es interessant ist
- Weil es interessant und lustig ist
- Weil ich gerne etwas am Computer mache (6 Mal)
- dass ich mich besser beim Computer auskenne (3 Mal)
- Weil ich es später vielleicht brauche (2 Mal)
- Weil es mir für die Zukunft (Beruf) bestimmt viel bringt und weil es Spaß macht
- Weil es cool ist (2 Mal)
- 7 Leermeldungen

Robot Carol und Gamemaker kombiniert mit Moodle eignen sich durch den hohen Motivationsgrad sehr gut für den Einstieg in der 5. Schulstufe.

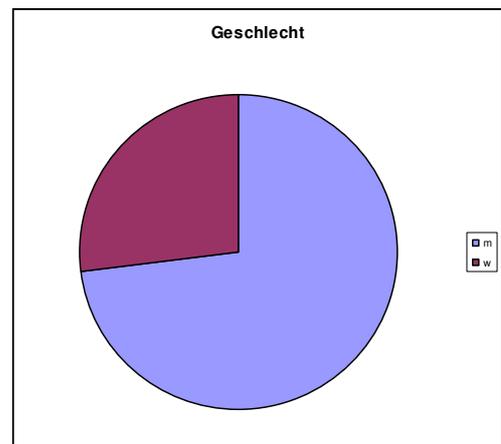
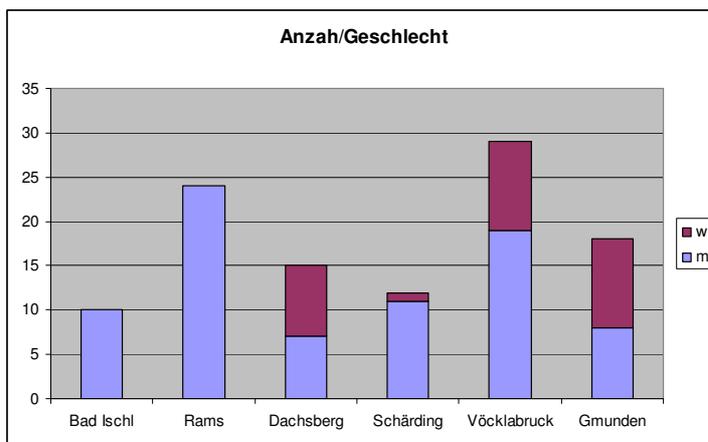
5 BEFRAGUNG DER SCHÜLERINNEN

Die Schülerinnen und Schüler wurden am Beginn des Projekts und am Ende des Projekts durch einen Fragebogen (siehe Anhang) über ihre Einstellung zum Problemlösung und im speziellen zum Programmieren befragt. Sie mussten außerdem Probleme (Denksportaufgaben) lösen.

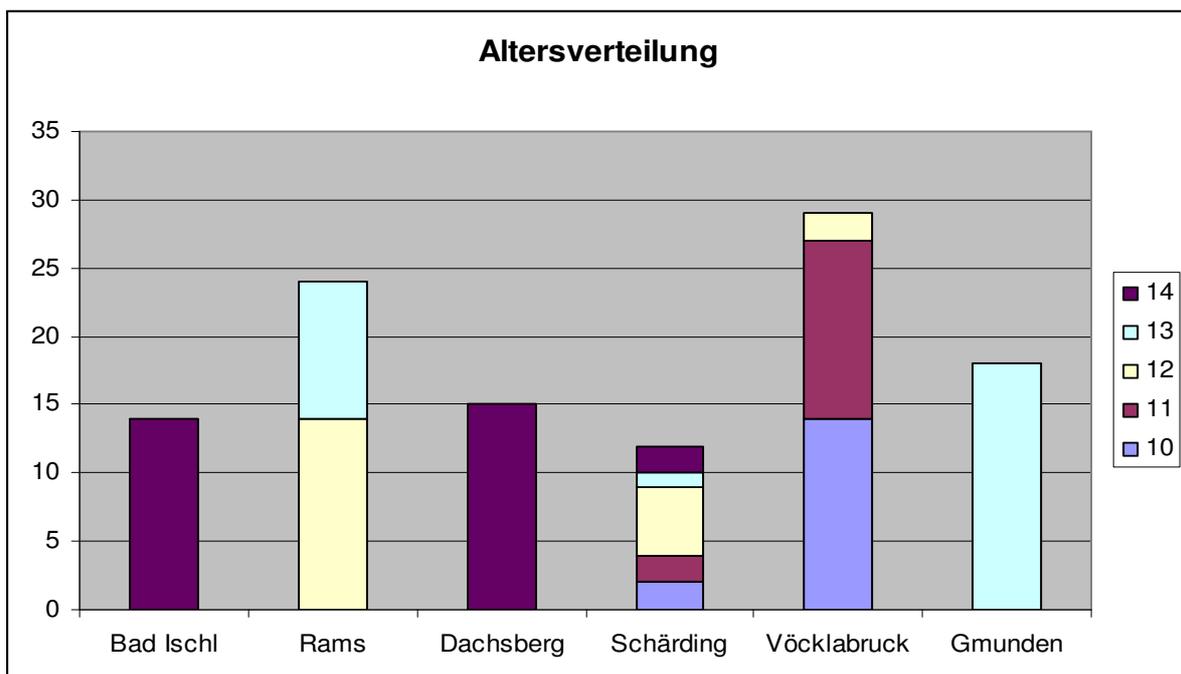
5.1 Ergebnisse

5.1.1 Zusammensetzung der Schülergruppen

Geschlecht



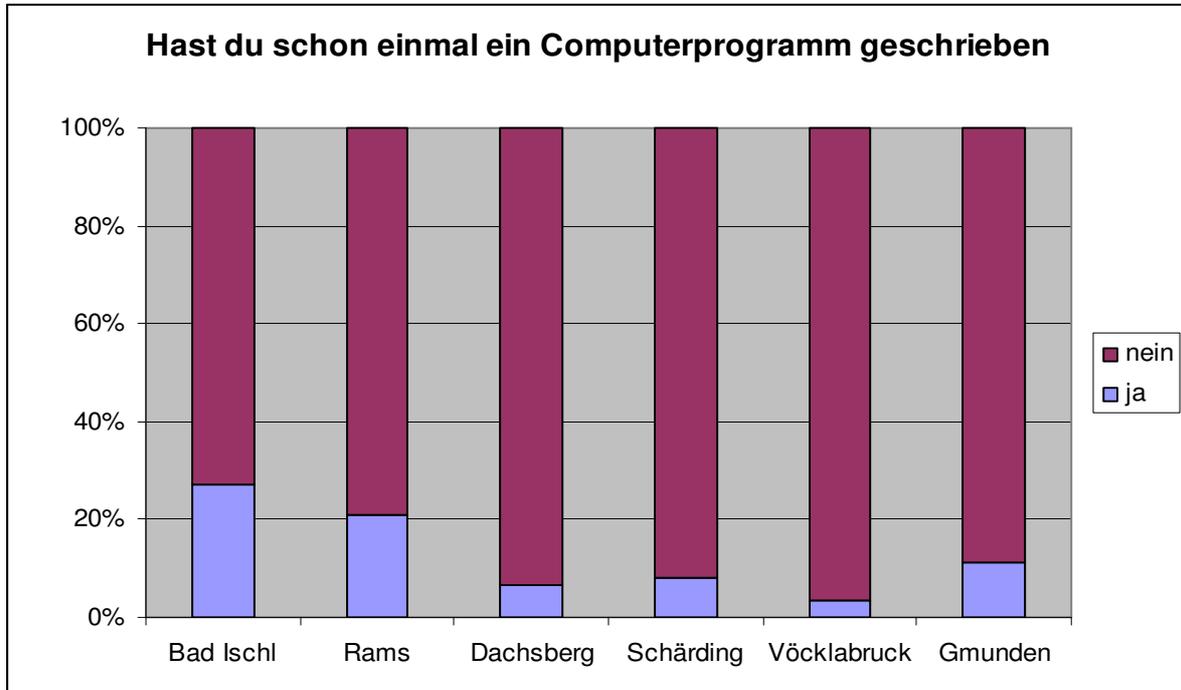
In Bad Ischl und im Ramsauergymnasium haben nur Knaben teilgenommen, in Schärding nur 1 Mädchen. In Gymnasium Dachsberg, Gmunden und Vöcklabruck war der Anteil der Mädchen höher. Insgesamt betrug der Anteil der Mädchen 27%, der Anteil der Knaben 73%



Die Altersverteilung wird in obigem Diagramm durch die verschiedenen Farben in den Säulen dargestellt. Das Alter der Schüler/innen lag zwischen 10 und 14 Jahren.

In Bad Ischl und Dachsberg kamen die Schüler aus einer 4. Klasse, in Gmunden und im Ramsauergymnasium aus einer Dritten Klasse in Vöcklabruck aus einer ersten Klasse und in Dachsberg war die Gruppe gemischt aus allen Klassen der Unterstufe.

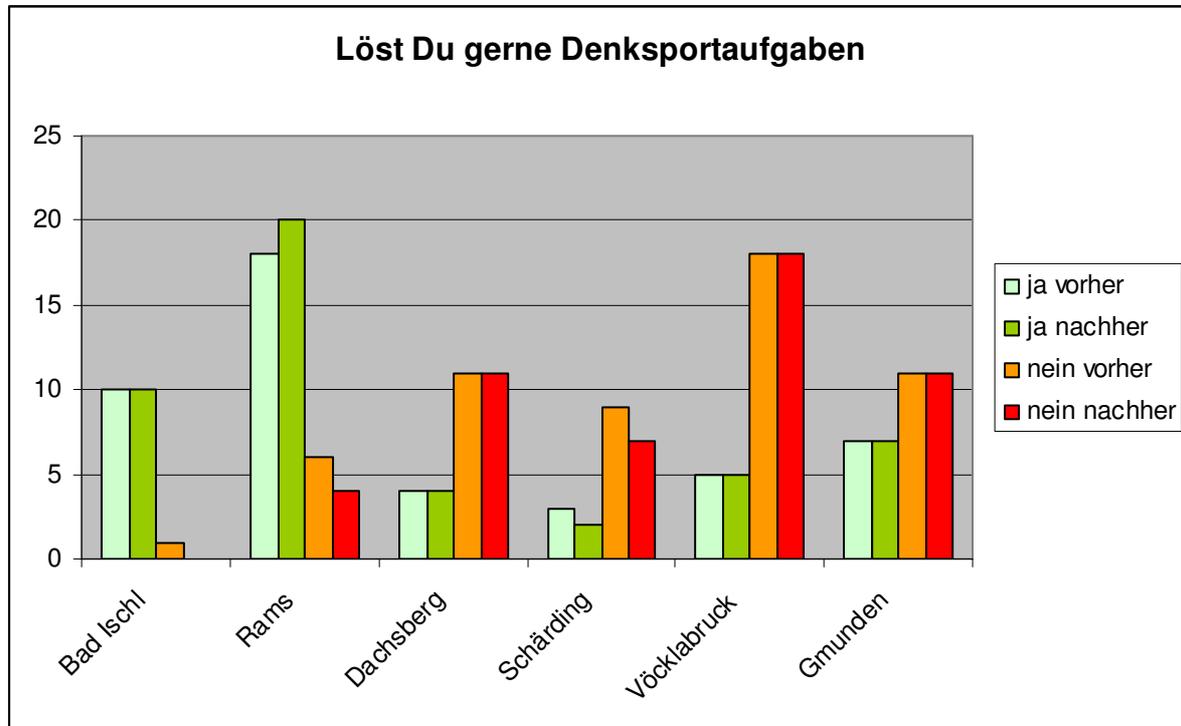
5.1.2 Programmiererfahrung



Programmiererfahrung hatten nur wenige Schüler/innen in den Programmiersprachen: Java, Visual Basic, Javascript, Java,

5.1.3 Veränderungen im Problemlöseverhalten in der Zeit des Programmierunterrichts

Löst du gerne Denksportaufgaben:

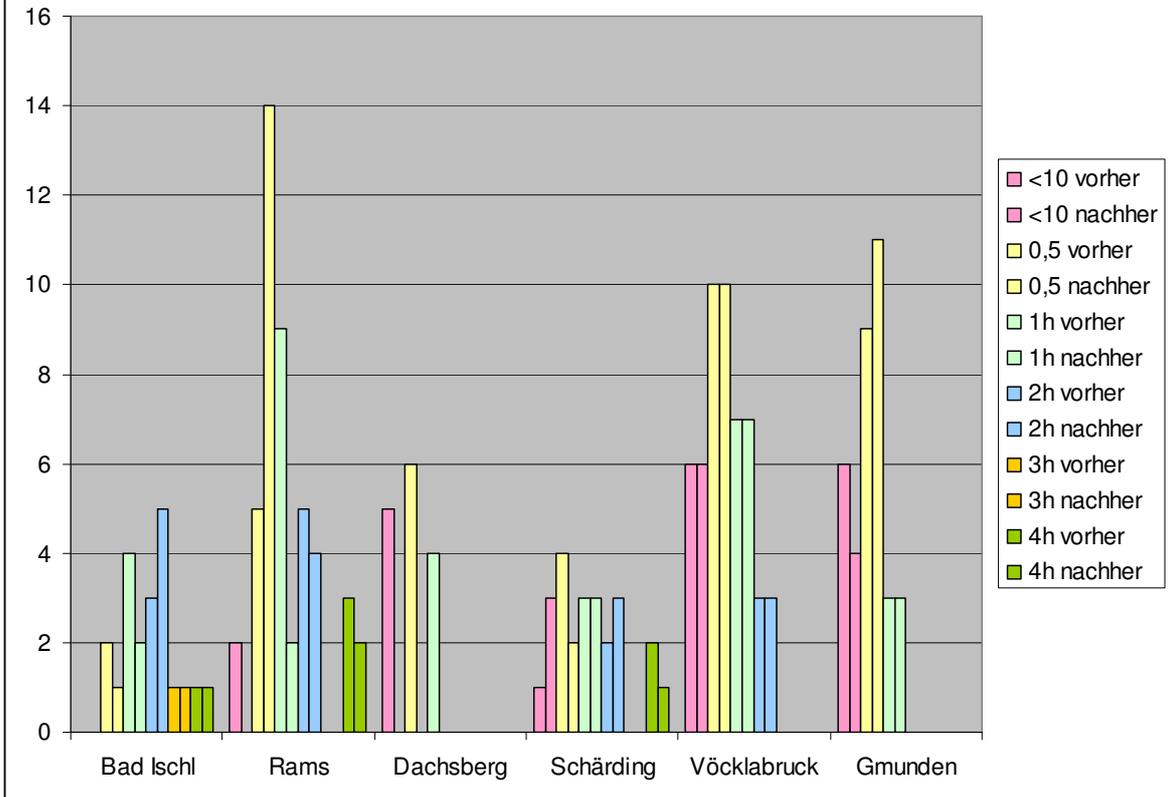


An der Freude am Lösen von Denksportaufgaben hat sich wenig verändert. Die Einstellungen dazu sind allerdings in den Schulen sehr unterschiedlich. Im Bad Ischl und im Ramsauergymnasium überwiegen die Schüler, die gerne solche Aufgaben lösen, in den anderen Schulen ist es umgekehrt.

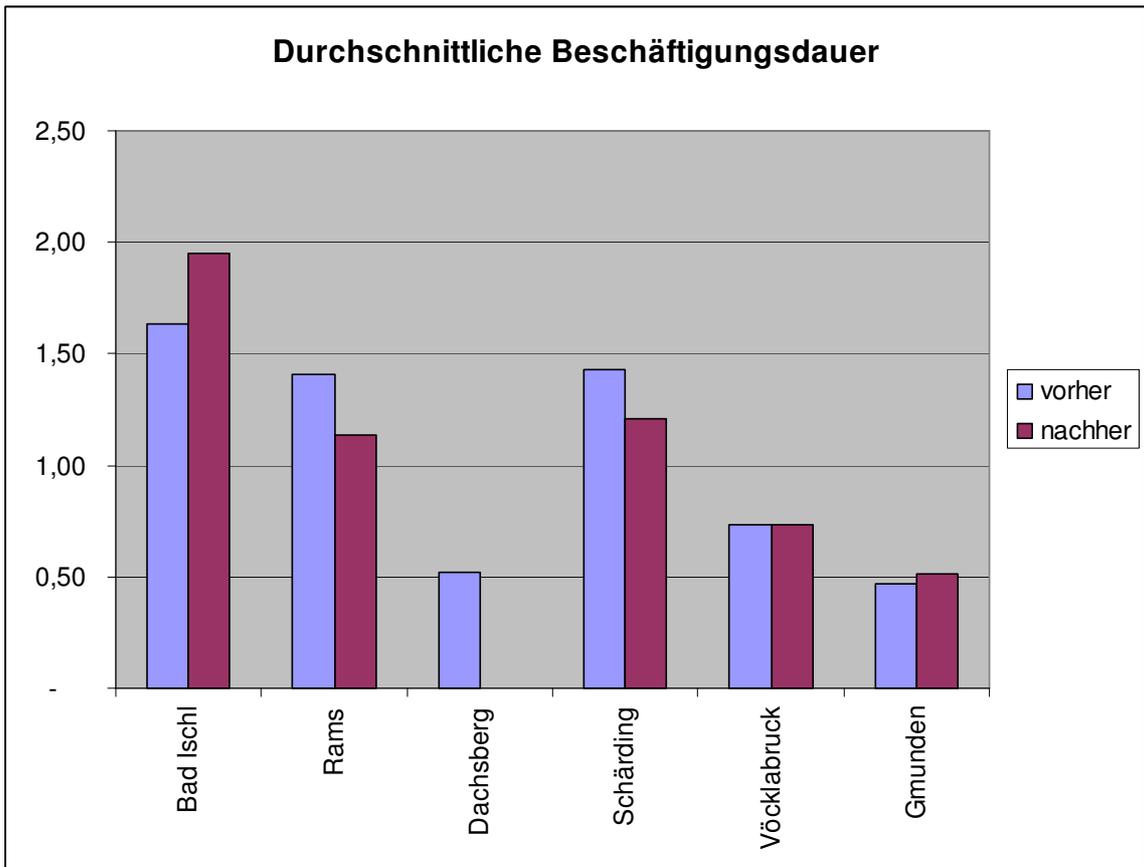
Wie lange kannst du dich mit einem Problem (Denksportaufgabe) beschäftigen?

Antwortmöglichkeiten: Weniger als 10 Minuten, 0,5 Stunden, 1 Stunde, ... 4 Stunden. Im Diagramm sind die Ergebnisse vor und nach dem Programmierunterricht einer Kategorie (z.B. weniger als 10 Minuten) in gleicher Farbe nebeneinander dargestellt. Die linke Säule gibt den Wert vorher und die rechte Säule den Wert nachher an. In Bad Ischl ist eine Zunahme der Schüler, die 2h an einem Problem arbeiten können, festzustellen. Diese haben vorher angegeben, sie könnten sich nur 0,5 bzw. 1h damit beschäftigen. Damit ergibt sich auch ein höherer Wert für die durchschnittliche Beschäftigungsdauer. Im Ramsauergymnasium und in Schärding ist eine Abnahme der durchschnittlichen Beschäftigungsdauer festzustellen. In Vöcklabruck sind alle Werte gleich geblieben. In Gmunden ist eine leichte Zunahme beobachtbar. Damit wage ich die Behauptung, dass ein Einfluss des Programmierunterrichts auf die Beschäftigungsdauer nicht festzustellen ist.

Ausdauer



Durchschnittliche Beschäftigungsdauer

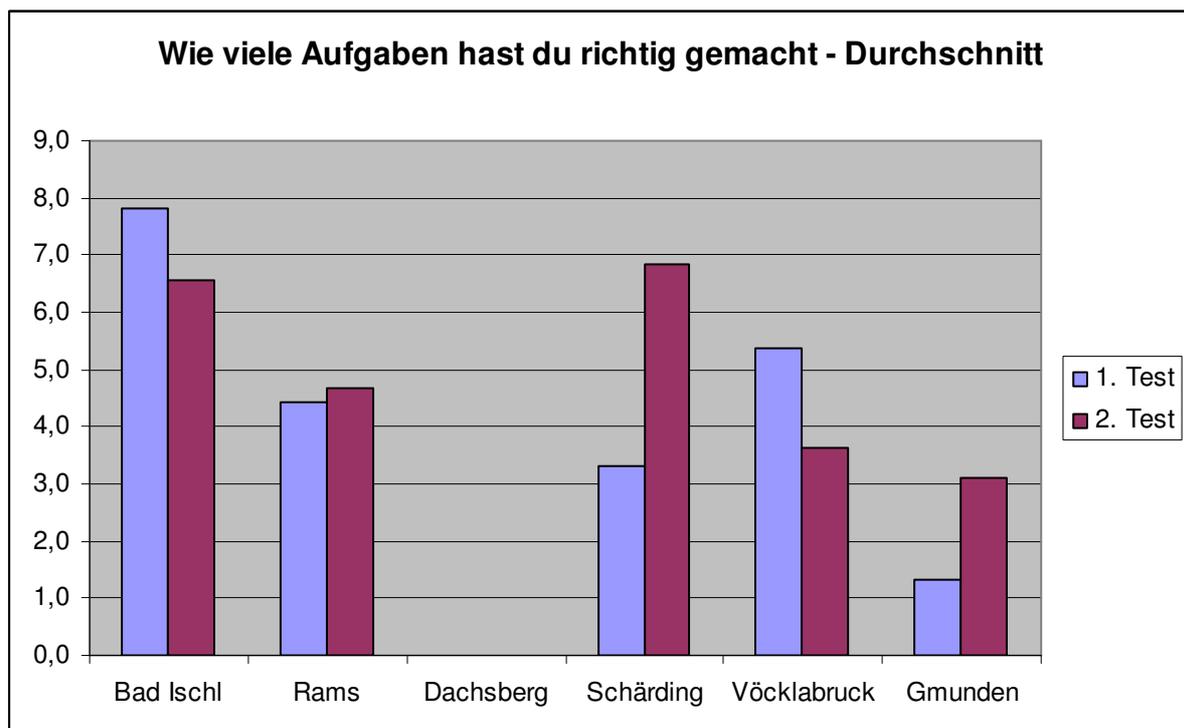
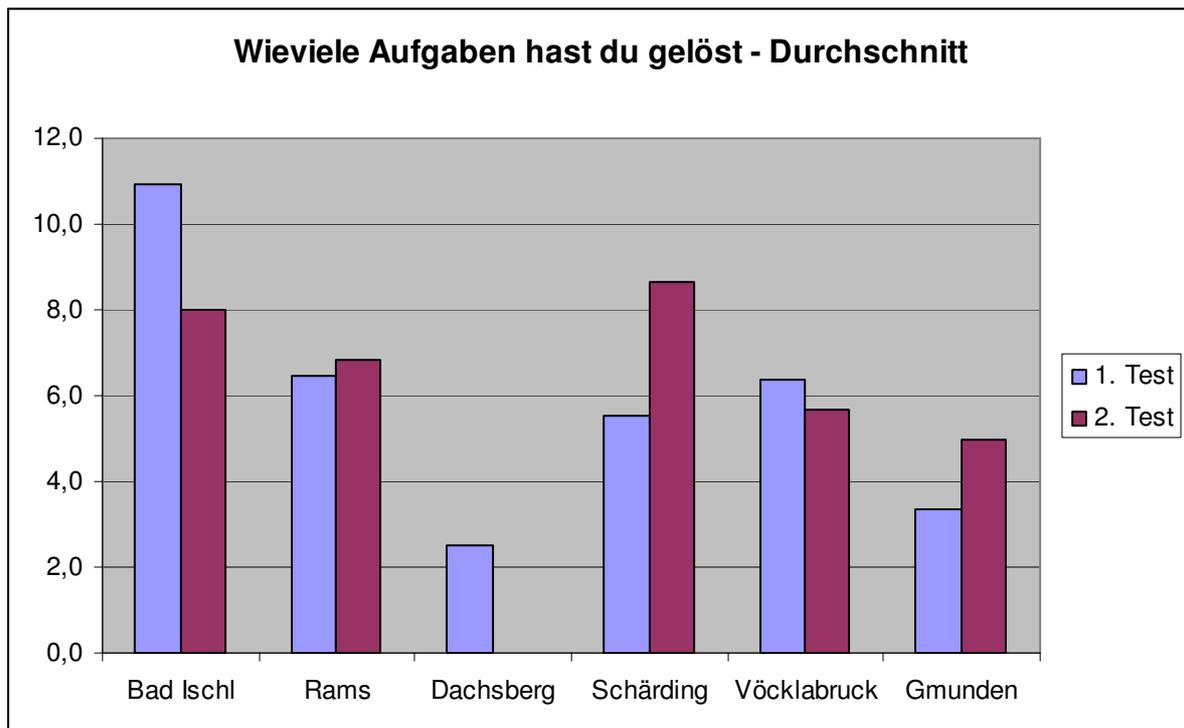


Wie viele Aufgaben hast du gelöst – Durchschnitt

Auch hier ist das Ergebnis sehr unterschiedlich. In Bad Ischl und Vöcklabruck haben die Schüler/innen beim zweiten Test weniger Aufgaben gemacht, im Ramsauergymnasium, in Schärding und Gmunden mehr Aufgaben.

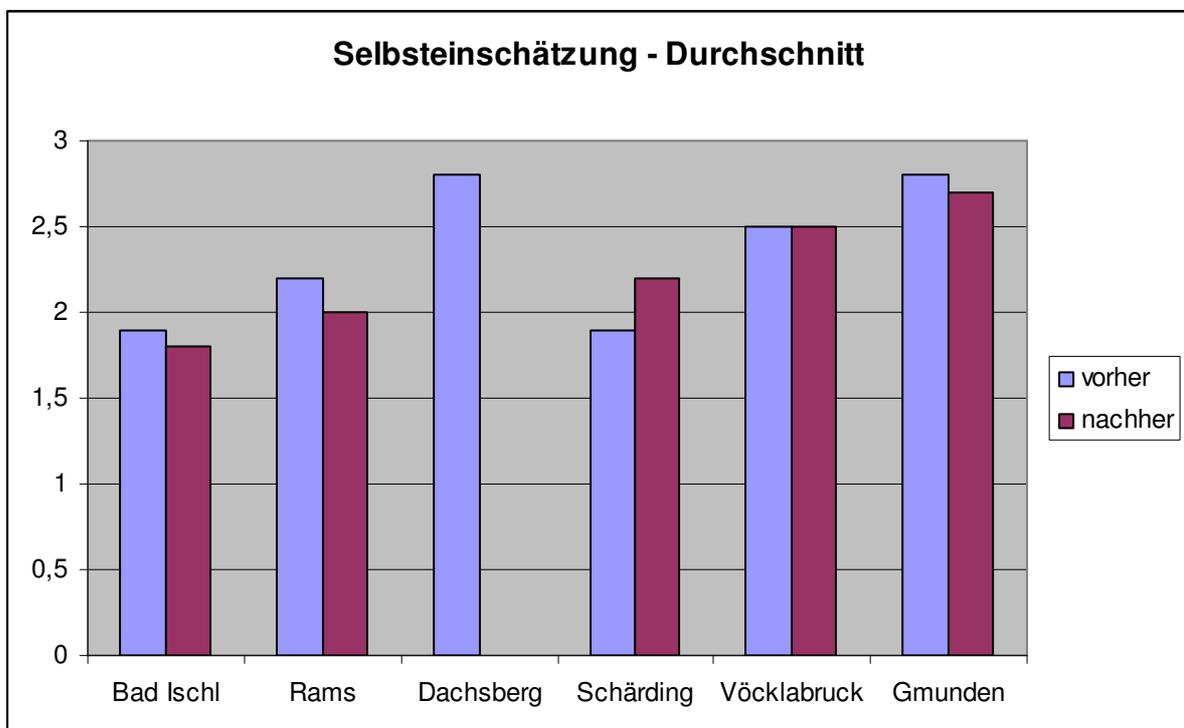
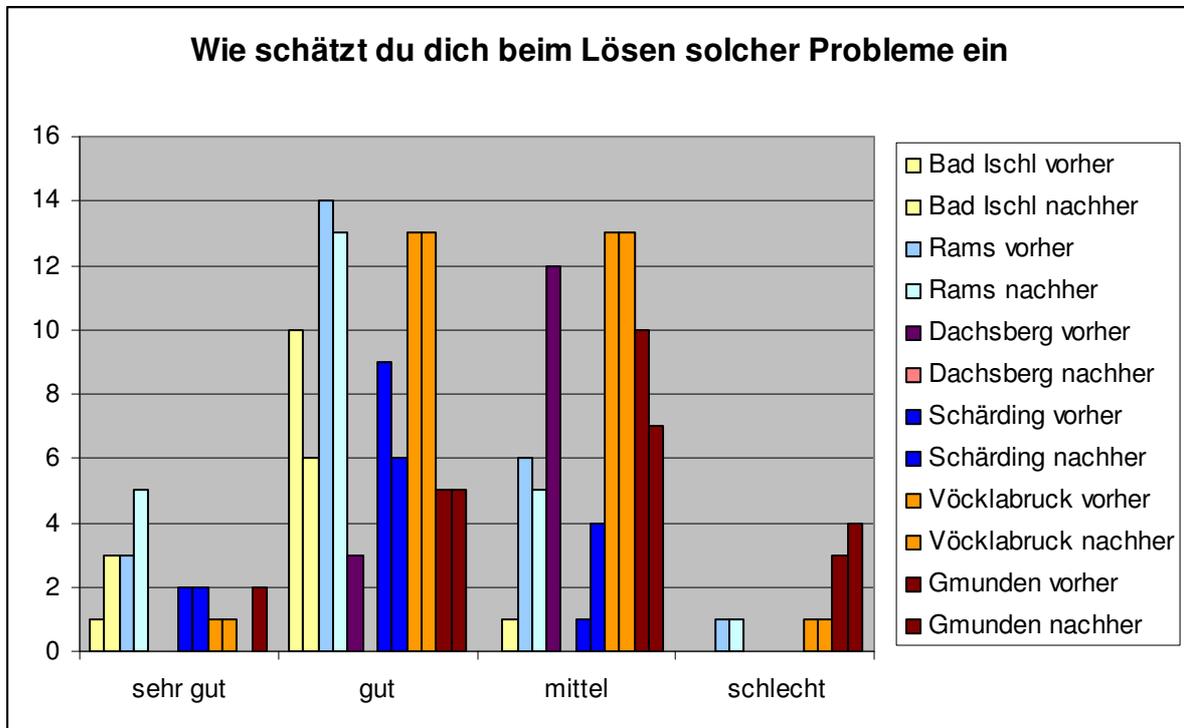
Wie viele Aufgaben hast du richtig gemacht – Durchschnitt:

Wiederum in Bad Ischl und Vöcklabruck beim zweiten Test weniger, die anderen Schulen mehr.



Wie schätzt du dich beim Lösen solcher Probleme ein?

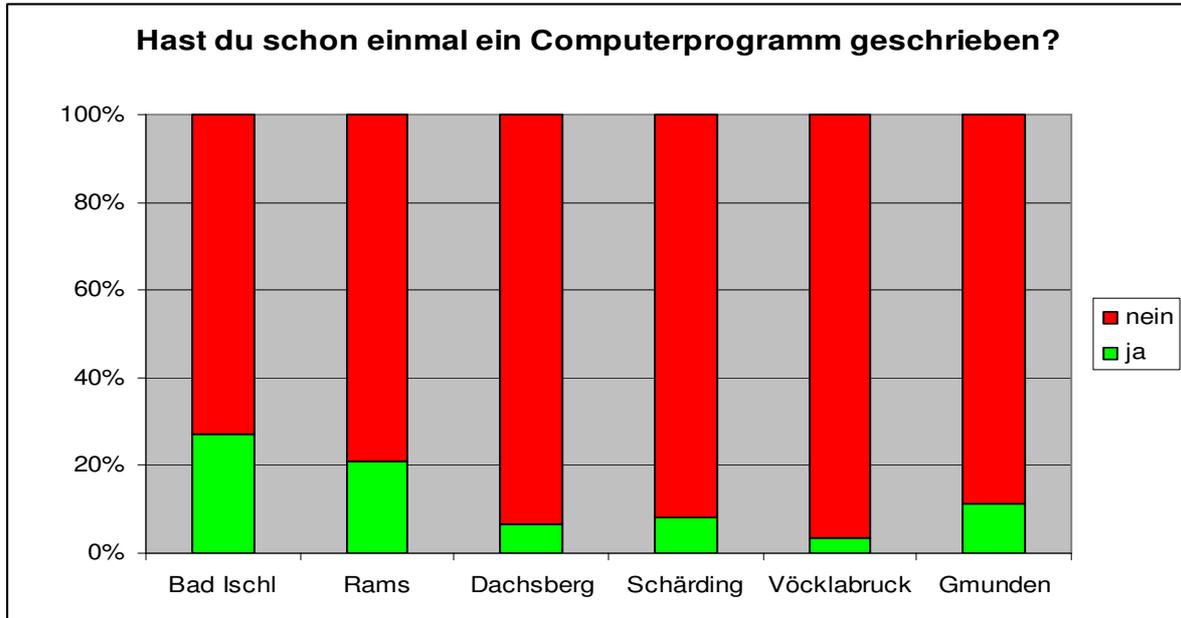
Bis auf Schärding schätzen sich im Durchschnitt die SchülerInnen nach dem Programmierunterricht etwas besser ein. (Notendurchschnitt: 1.. sehr gut, 2..gut, 3..mittel, 4..schlecht)



5.1.4 Einstellung zum Programmieren und zum Programmierunterricht

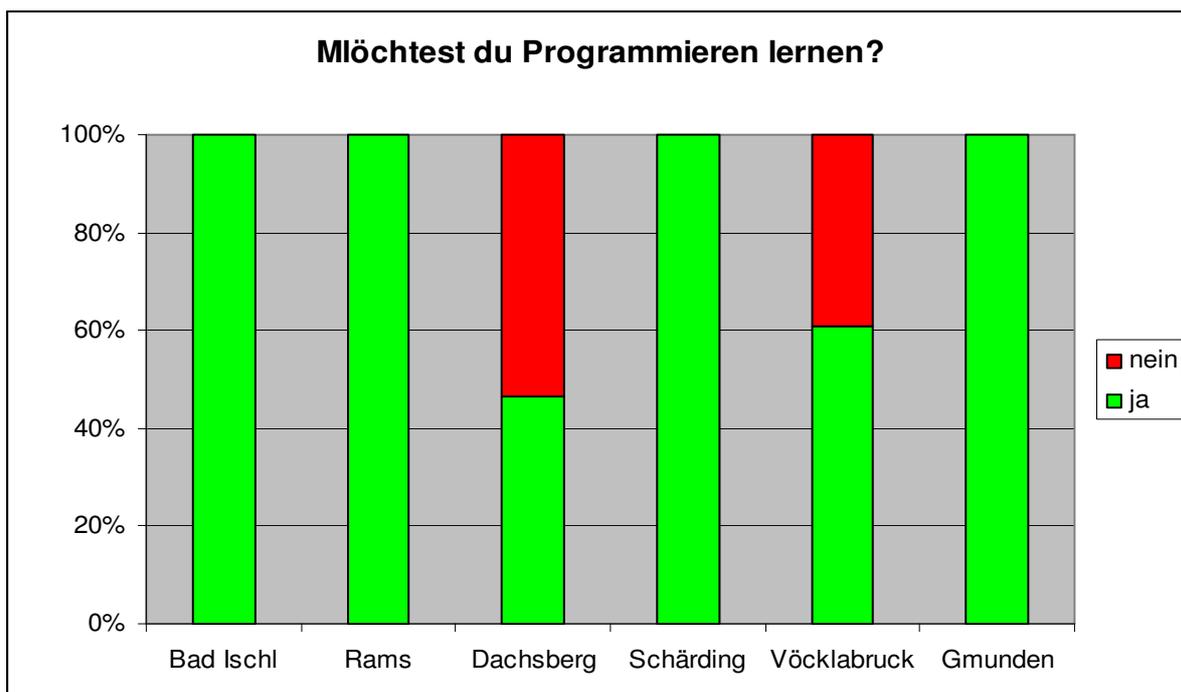
Hast du schon einmal ein Computerprogramm geschrieben?

Das Ergebnis der Befragung zeigt, dass nur sehr wenige Schüler/innen Programmiererfahrung hatten. Der höchste Wert in Bad Ischl (25%).



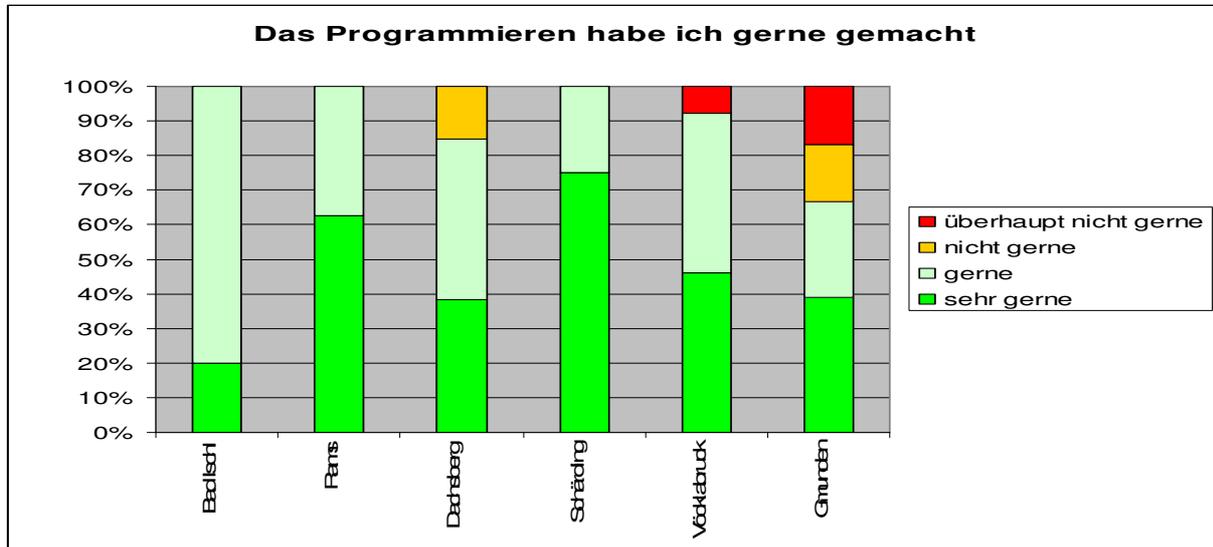
Möchtest du Programmieren lernen?

Alle Schüler/innen in Bad Ischl, Ramsauergymnasium, Schärding und Gmunden wollten zu Beginn Programmieren lernen. Nur in Dachsberg (50%) und Vöcklabruck (40%) nicht.



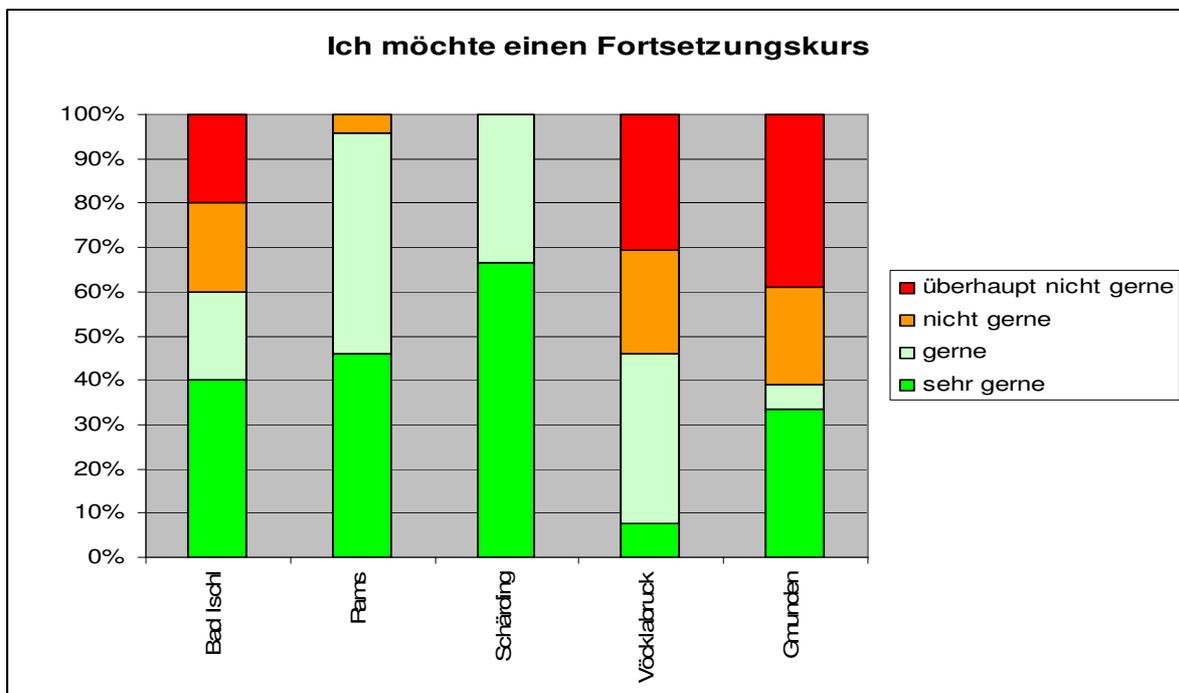
Das Programmieren habe ich gerne gemacht.

Es gab nur ganz wenige Schüler/innen, die das Programmieren nicht gerne oder überhaupt nicht gerne gemacht haben (Vöcklabruck, Gmunden). Das beste Ergebnis ist in Schärding festzustellen (75% sehr gerne).



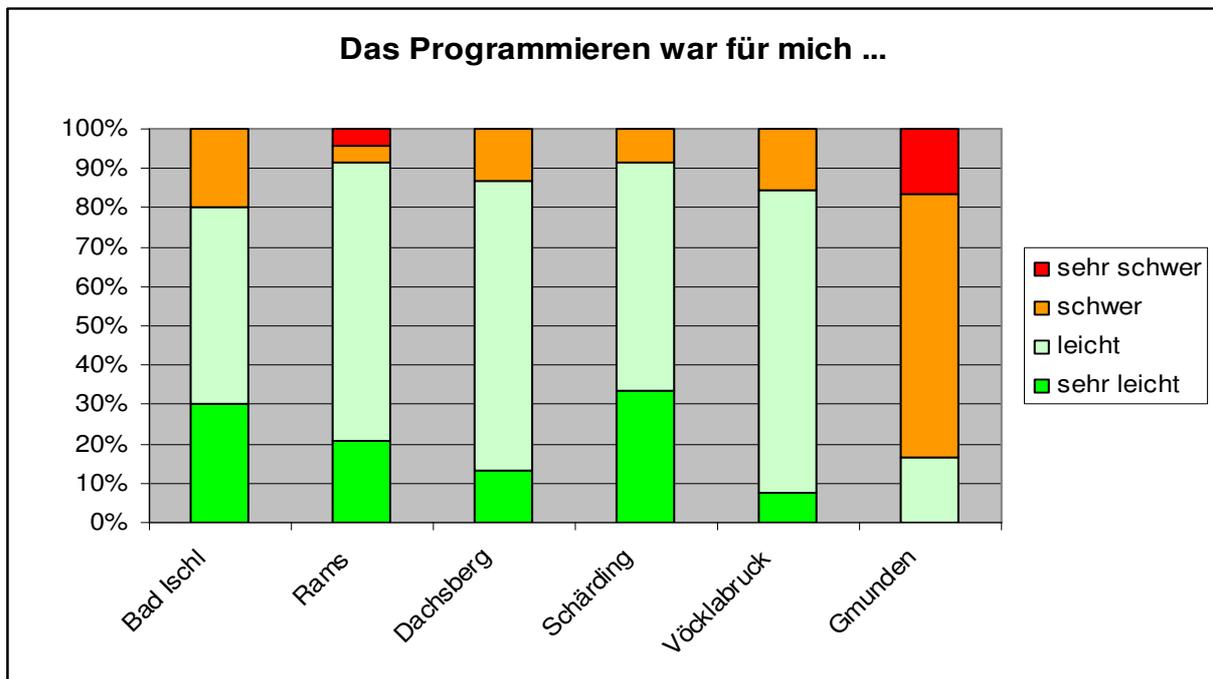
Ich möchte einen Fortsetzungskurs

In Schärding wollen alle Schüler/innen einen Fortsetzungskurs, im Ramsauergymnasium fast alle, in Bad Ischl 60%, in Vöcklabruck 45% und in Gmunden ca. 40%. Dazu ist zu bemerken, dass in Vöcklabruck und in Gmunden alle SchülerInnen einer Klasse diesen Unterricht besucht haben und in den anderen Schulen nur Freiwillige.



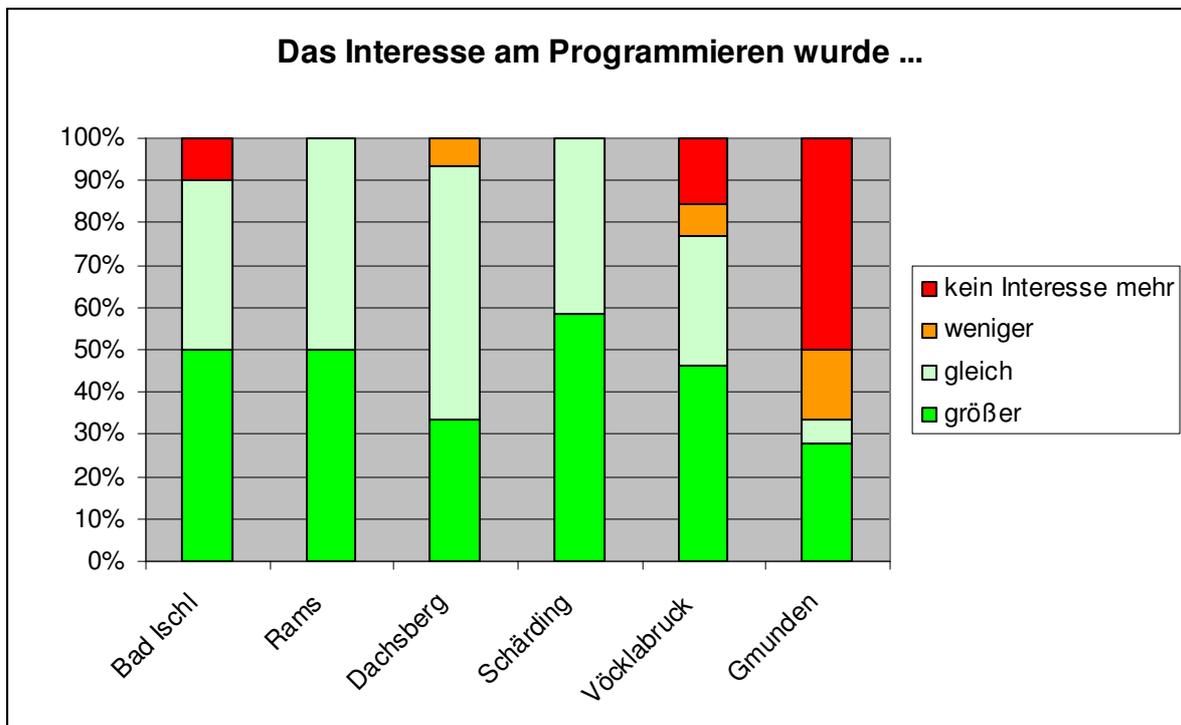
Das Programmieren war für mich ...

Nur in Gmunden haben die Schüler/innen das Programmieren eher „schwer“ eingeschätzt. In allen anderen Schulen fanden es die Schüler/inne „sehr leicht“ bis „leicht“.



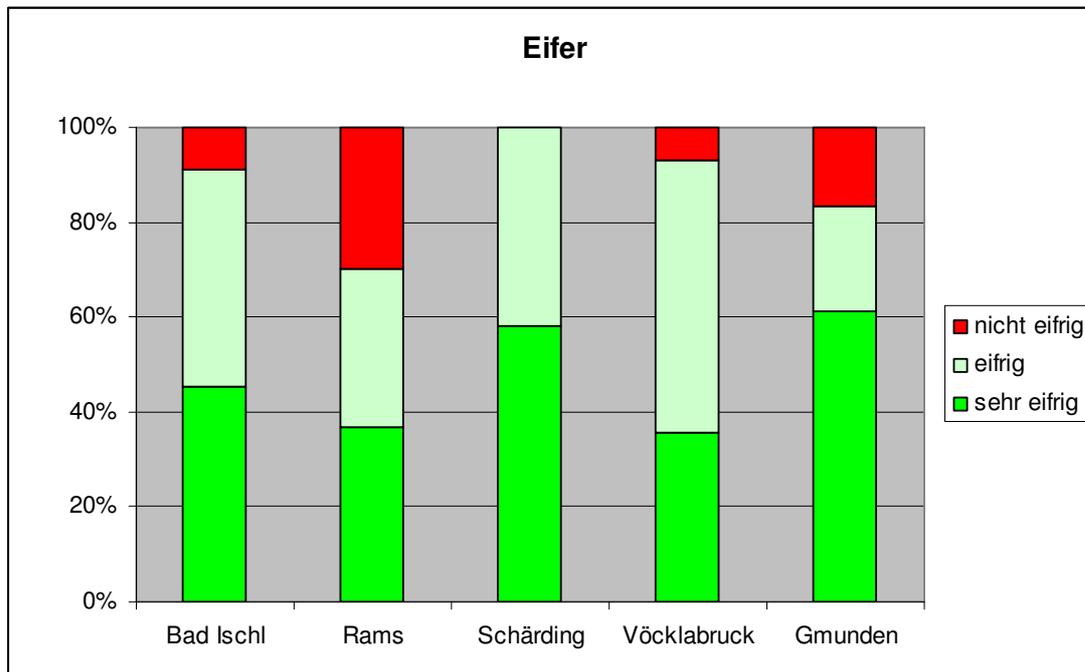
Das Interesse am Programmieren wurde/ist

Das Interesse am Programmieren wurde bei den meisten Schüler/innen größer oder ist gleich geblieben. Eine Ausnahme stellt Gmunden dar. Dort haben 50% der Schüler/innen kein Interesse am Programmieren mehr, obwohl zu Beginn alle Schüler/innen programmieren lernen wollten.



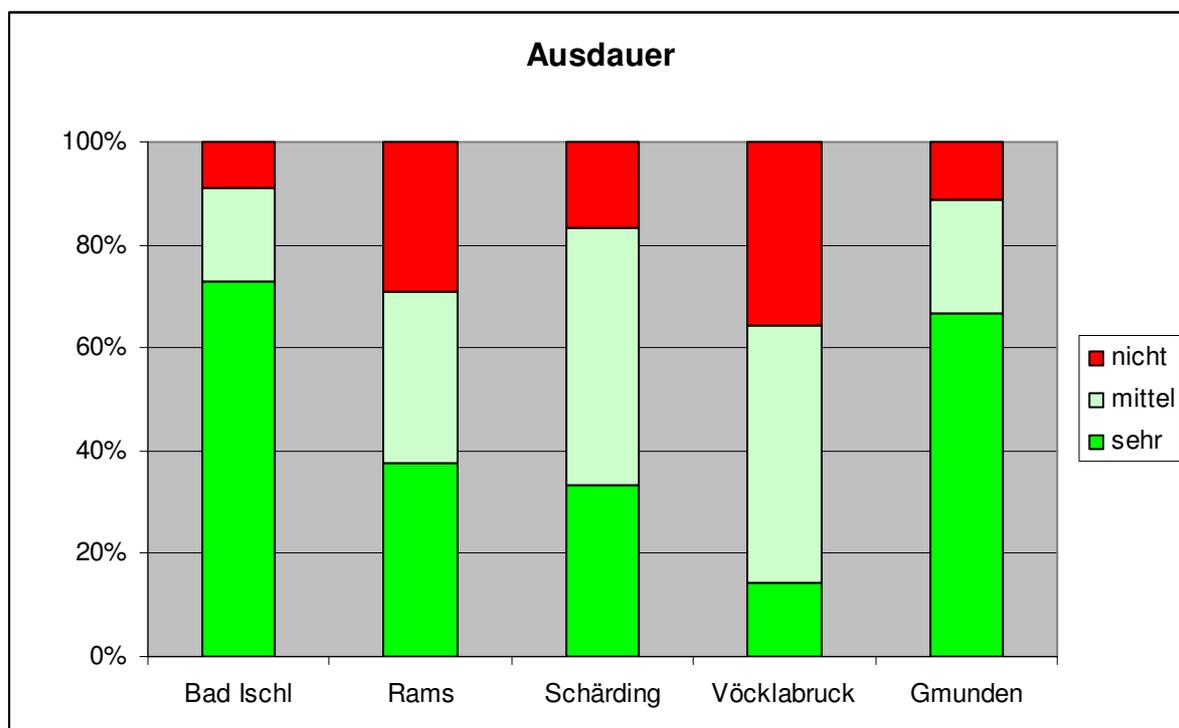
5.1.5 Einschätzung durch die Lehrerinnen und Lehrer

Die Lehrer/innen schätzen die Schüler sehr eifrig ein. Nur im Ramsauergymnasium werden 30% der SchülerInnen nicht eifrig eingestuft.

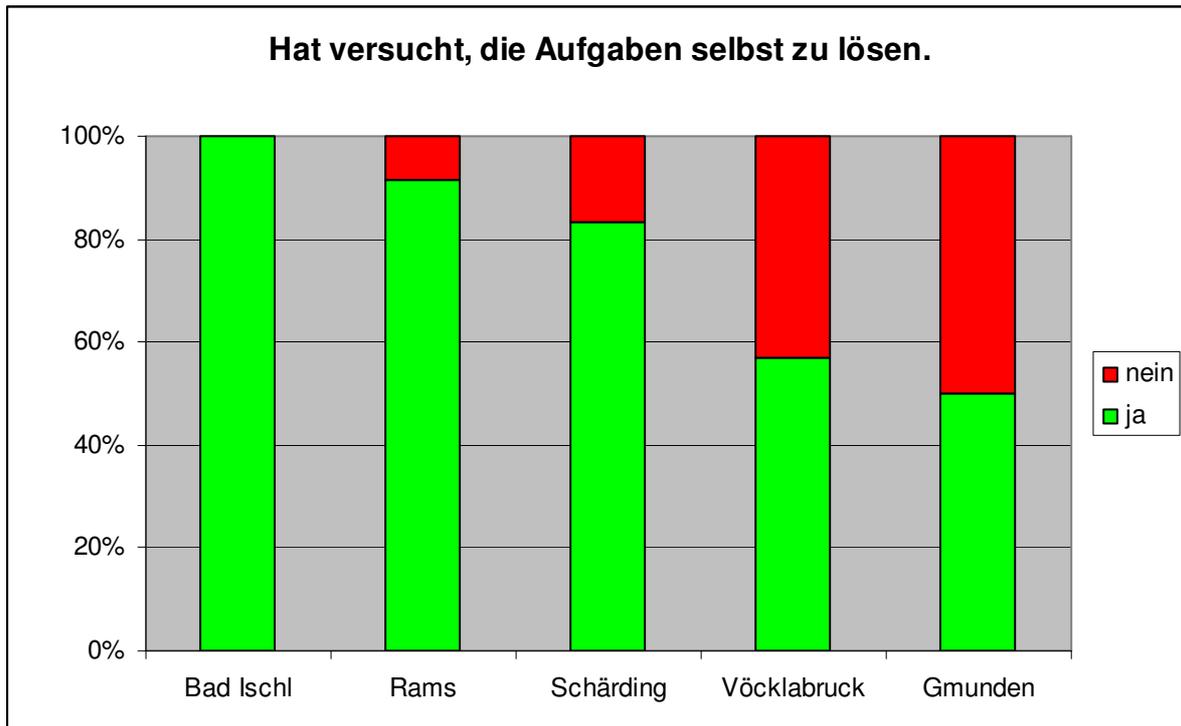


Ausdauer

70% - 90% der Schüler/innen werden ausdauernd eingestuft. Das beste Ergebnis kommt aus Bad Ischl, das schlechteste aus Vöcklabruck.

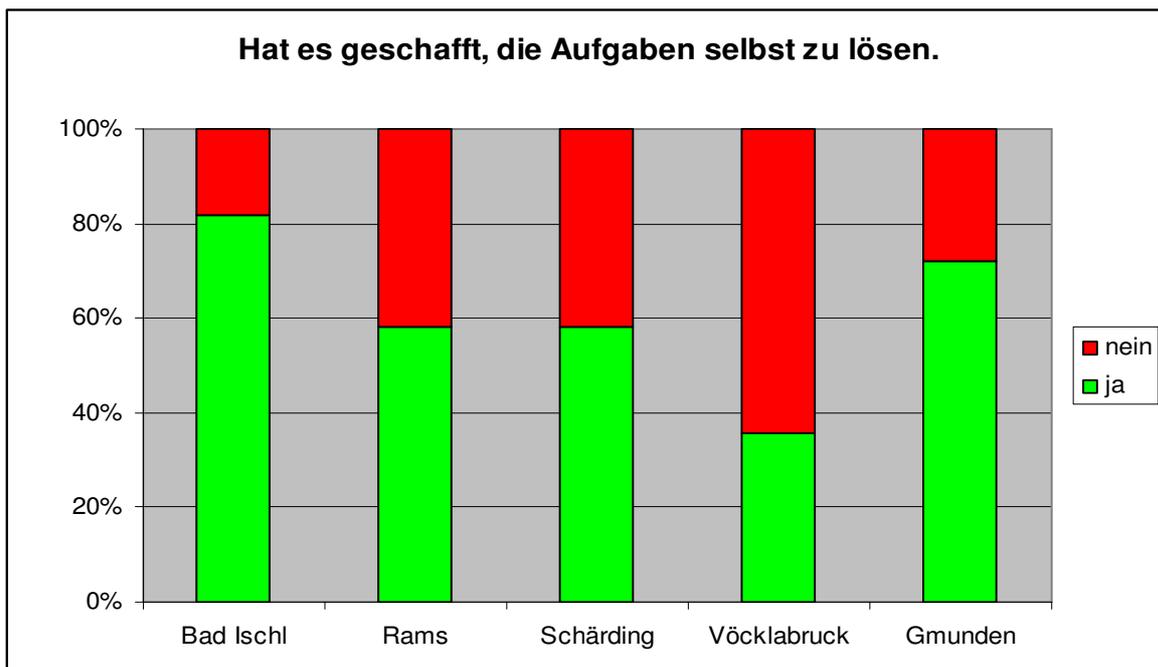


Hat versucht, die Aufgaben selbst zu lösen,
wird von 100% bis 50% eingeschätzt.



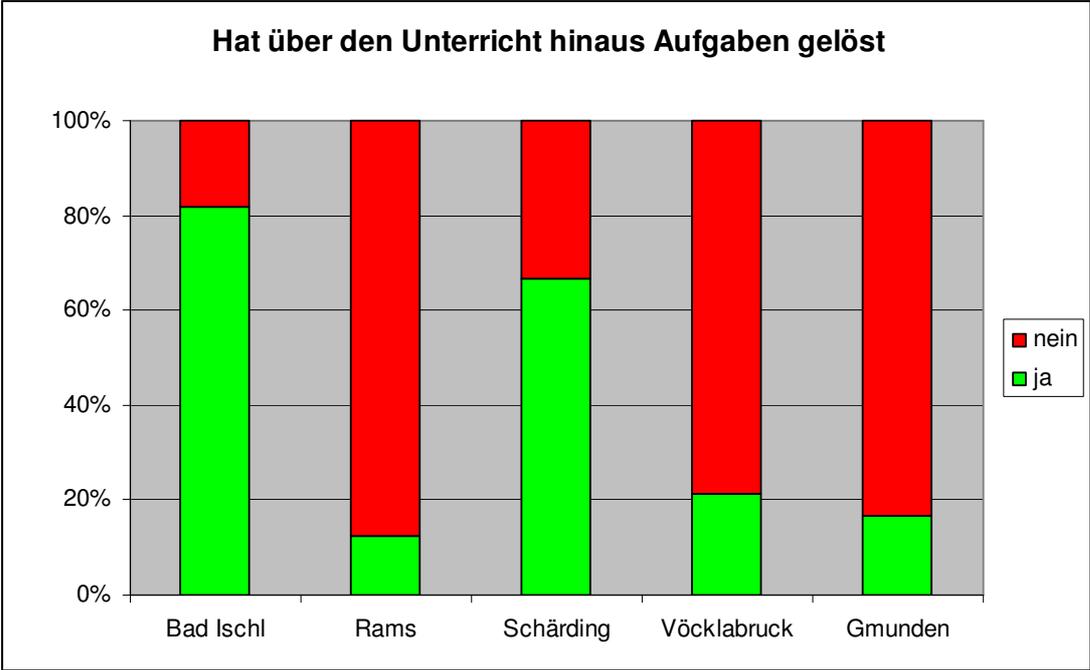
Hat es geschafft die Aufgaben selbst zu lösen

Wie aus der Beantwortung der obigen Frage hervorgeht, haben zwar die meisten Schüler/inne versucht, die Aufgaben selbst zu lösen, aber nicht im gleichen Ausmaß es geschafft. Der Wert für „geschafft“ geht von 80% (Bad Ischl) bis 35% (Vöcklabruck). Hier dürfte auch das Alter der SchülerInnen eine Rolle spielen. In Vöcklabruck waren es durchwegs SchülerInnen einer ersten Klasse, in Bad Ischl kamen die SchülerInnen aus der 4. Klasse.



Hat über den Unterricht hinaus Aufgaben gelöst:

Bemerkenswert ist der hohe Anteil in Bad Ischl und Schärding.



6 ZUSAMMENFASSUNG

6.1 Programmierunterricht in der Unterstufe

Programmierunterricht in der Unterstufe auf freiwilliger Basis (Freigegegenstand, Unverbindliche Übung) hat sich bewährt. Es hatten vor der Untersuchung nur wenige Schülerinnen und Schüler Programmiererfahrung und es wollten fast alle, die teilgenommen haben, das Programmieren lernen. Mit wenigen Ausnahmen haben die Schülerinnen und Schüler das Programmieren gerne gemacht und sie haben es auch nicht schwer gefunden. Das Interesse am Programmieren ist gestiegen und die meisten sind an einem Fortsetzungskurs interessiert.

Für viele Schüler/innen der 1. Klasse waren die Aufgaben teilweise zu schwierig. Ein Einsatz in der 3. und 4. Klasse ist besser. Klassenübergreifende Kurse von der 1. bis zur 4. Klasse haben sich nicht bewährt.

Aufgaben, welche der Kreativität der Schülerinnen und Schüler mehr Raum geben, kommen besser an, als Aufgaben, die sehr einschränken. Freiere Aufgabenstellungen sind motivierender.

6.2 Einsatz von Programmierlernumgebungen

Der Einsatz der Programmierumgebungen hat sich bewährt, allerdings nur zur Einführung in die Programmierung (6-8 Doppelstunden). Bei Robot Karol wurde das Fehlen von Variablen bemängelt. Einfache Beispiele lösten die Schüler gerne und erfolgreich, zur Lösung von komplexeren Aufgaben konnten viele jedoch nicht motiviert werden.

Java Kara wurde in einigen Schulen erfolgreich eingesetzt, allerdings nur die Programmierung als Automat. Die Programmierung in Java dürfte wegen der komplexen Syntax (vor allem wegen der Klammersetzung) für die Unterstufe zu schwierig sein.

Nach einigen Doppelstunden der Programmierung von Robot Karol sank das Interesse der Schülerinnen. Durch de Einsatz von LOGO, Java Kara oder Gamemaker konnte es aber wieder geweckt werden. Insbesondere durch die Turtlegrafik konnten die Schülerinnen und Schüler wieder motiviert werden.

Für viele Schüler/innen der 1. Klasse waren die Aufgaben teilweise zu schwierig. Ein Einsatz in der 3. und 4. Klasse ist besser. Klassenübergreifende Kurse von der 1. bis zur 4. Klasse haben sich nicht bewährt.

Aufgaben, welche der Kreativität der Schülerinnen und Schüler mehr Raum geben, kommen besser an, als Aufgaben, die sehr einschränken. Freiere Aufgabenstellungen sind motivierender.

6.3 Einfluss auf die Problemlösekompetenz

Die unterschiedlichen Ergebnisse der beteiligten Schulen lassen keinen Schluss zu, dass sich der Programmierunterricht auf die Problemlösekompetenz der Schülerinnen und Schüler ausgewirkt hätte. Die Veränderungen sind sehr gering und widersprüchlich. In einer vierstufigen Skala schätzten sich die Schülerinnen und Schüler im Durchschnitt von 1,7 bis 2,7 ein. In 3 Schulen schätzten sie sich nach dem Pro-

grammierunterricht besser ein, in einer schlechter und in einer gleich gut. Aus zwei Schulen gab es keine Vergleichswerte.

Die eigenständige Lösung der Programmieraufgaben schafften in Bad Ischl 80% und Vöcklabruck nur 36%, die anderen Schulen lagen die Werte um die 60%. Hier dürfte sich das Alter der Schüler/innen ausgewirkt haben. In Vöcklabruck, wo dieser Wert eher niedrig war, kamen die Schüler/innen aus einer ersten Klasse und in Bad Ischl, wo der Wert hoch war, aus der 4. Klasse.

Die durchschnittliche Anzahl der richtig gelösten Aufgaben ist in 3 Schulen gestiegen, in 2 Schulen gesunken und war in den verschiedenen Schulen sehr unterschiedlich (Maximalwert 7.8, Minimalwert: 1.3).

6.4 Einstellung und Arbeitshaltung der Schülerinnen und Schüler

Bei allen Fragen, die Ausdauer oder Erfolg beim Problemlösen betrafen, gab es gegensätzliche Ergebnisse. Bei einigen Schulen war eine Verbesserung zu sehen bei einigen eine Verschlechterung.

Die durchschnittliche Beschäftigungsdauer reichte von ca. 1,5 Stunden bis 0,5 Stunden. In 3 Schulen ist diese gestiegen oder gleich geblieben, in zwei Schulen gesunken. Aus 2 Schulen gab es keine Vergleichswerte.

Die Freude am Lösen von Denksportaufgaben hat sich kaum verändert, war aber in den verschiedenen Schulen sehr unterschiedlich (Bad Ischl gaben beim 2. Test alle an, dass sie solche Aufgaben gerne lösen, in Schärding beim ersten Test nur 2 Schüler)

Die Anzahl der gelösten Denksportaufgaben ist in 3 Schulen gestiegen, in 2 Schulen gesunken (aus 2 gab es keine Vergleichswerte) und war in den verschiedenen Schulen auch sehr unterschiedlich (Maximalwert 11, Minimalwert: 3).

Die Einschätzung der Lehrpersonen ergab, dass 70% bis 100% mit Eifer dabei waren.

6.5 Einstellung zum Programmieren und zum Programmierunterricht

Vor dem Programmierunterricht hatten nur wenige Schüler/innen Programmiererfahrung. In 4 Schulen gaben alle Schüler/innen an, programmieren lernen zu wollen. In Schulen, wo der Unterricht auf freiwilliger Basis durchgeführt wurde (als unverbindliche Übung oder Kurs) gaben fast alle Schüler/innen an, dass sie das Programmieren sehr gerne oder gerne gemacht haben. In Schulen mit Pflichtgegenstand gab es einige, die es nicht gerne machten (Dachsberg 15%, Gmunden 25%).

Die Anzahl der Schüler/innen, die einen Fortsetzungskurs wünschen, war in zwei Schulen sehr hoch (95%, 100%), in einer Schule bei 60% und in 2 Schulen bei 40%.

In den meisten Schulen (5) meinten die Schüler/innen, dass ihnen das Programmieren „sehr leicht“ bzw. „leicht“ gefallen sei, nur in einer Schule lag der Wert bei 15%. Dort wurde zum Großteil Visual Basic und nicht die Programmierumgebung Robot Karol bzw. JavaKara eingesetzt. Aus meiner Sicht zeigt dieses Ergebnis, dass zum Einstieg in die Programmierung die einfacheren Programmierumgebungen besser geeignet sind.

Die Frage, ob das Interesse am Programmieren größer geworden sei, wurde in ähnlicher Weise beantwortet. In 5 Schulen wurde das von einer großen Mehrheit (75% bis 100%) bejaht, in der einen Schule nur von 34%.

7 SCHLUSSFOLGERUNGEN UND AUSBLICK

Die eingesetzten Programmierumgebungen eignen sich sehr gut für den Einstieg in das Programmieren in der Unterstufe. Diese können von der ersten Klasse bis zur 4. Klasse sinnvoll verwendet werden. Das Abstraktionsvermögen und die Fähigkeit zur eigenständigen Lösung von Aufgaben nehmen mit dem Alter zu. Bei den älteren Schüler/innen haben 60% bis 80% angegeben, dass sie es geschafft haben, die Aufgaben selbst zu lösen, bei den 10jährigen nur 35%.

Das Interesse am Programmierunterricht ist bei Unterstufenschülern vorhanden und nimmt beim Einsatz von Robot Karol, Java Kara und Logo zu. In einem Pflichtgegenstand mit Visual Basic ist das Interesse geringer, die Programmierung wird als schwer empfunden und eine Fortsetzung eher nicht gewünscht.

Diese Ergebnisse lassen für mich den Schluss zu, dass in der Unterstufe in allen Schulen Unverbindliche Übungen oder Kurzkurse angeboten werden sollten, in denen die Schüler/innen mit Programmierumgebungen wie RobotKarol, JavaKara oder Logo das Programmieren lernen können. Das Angebot sollte eher in der 3. und 4. Klasse gemacht werden.

Ein Einfluss des Programmierunterrichts auf die Problemlösekompetenz und das Problemlöseverhalten kann aus der Untersuchung nicht abgeleitet werden. Die Ergebnisse in den Schulen sind sehr unterschiedlich und zeigen keinen eindeutigen Trend.

8 LITERATUR

GRABINGER B. (1993). Arbeitsbuch LOGO. Bonn: Dümmler.

HARTMANN Werner (2006). Informatikunterricht planen und durchführen. Berlin, Heidelberg, New York: Springer.

REICHERT Raimond, NIEVERGELT Jürg, HARTMANN Werner (2005). Programmieren mit Kara. 2. Auflage, Berlin, Heidelberg, New York: Springer.

Internetadressen:

Robot Karol

<http://www.schule.bayern.de/karol/>

http://ods.schule.de/bics/inf2/didaktik/minilanguages/aufgaben/uebungsaufgaben_robot_karol.html

<http://www.lag-informatik.ksn.at/programmieren.html>

<http://www.luise-buechner-schule.eisfair.net/moodle/course/view.php?id=50>

JAVA KARA

<http://www.swisseduc.ch/informatik/karatojava/>

<http://www.oberstufeninformatik.de/info11/javakara.html>

<http://www.infosense.ch/reichert/docs/infos2003.pdf>

LOGO

<http://www.softronix.com/logo.html>

<http://sourceforge.net/projects/fmslogo>

ANHANG

Der Fragebogen vor dem Programmierunterricht

Fragebogen am Beginn des Projekts „Programmieren Unterstufe“

Allgemeiner Teil

<i>Frage</i>	<i>Antwort</i>
Geschlecht	<input type="checkbox"/> männlich <input type="checkbox"/> weiblich
Alter	
Klasse	
Hast Du schon einmal ein Computerprogramm geschrieben?	<input type="checkbox"/> ja <input type="checkbox"/> nein
Wenn ja: In welcher Programmiersprache	
Wenn ja: Versuche zu erklären, was Programmieren ist	
Spielst du gerne mit dem Computer	<input type="checkbox"/> ja <input type="checkbox"/> nein
Welche Spiele spielst du?	
Spielst du Schach oder Ähnliches	<input type="checkbox"/> ja <input type="checkbox"/> nein
Löst du gerne Denksportaufgaben	<input type="checkbox"/> sehr gerne <input type="checkbox"/> gerne <input type="checkbox"/> nicht gerne
Löst du neue mathematische Aufgaben?	<input type="checkbox"/> ja <input type="checkbox"/> nein
Kommst du dabei selbst auf Lösungen	<input type="checkbox"/> ja <input type="checkbox"/> nein
Wie lange (Minuten, Stunden) kannst du dich mit einem Problem (Denksportaufgabe, Mathematik, ...) beschäftigen?	
Wie schätzt du dich beim Lösen solcher Probleme ein?	<input type="checkbox"/> sehr gut <input type="checkbox"/> gut <input type="checkbox"/> mittelmäßig <input type="checkbox"/> schlecht
Wenn dir die Lösung nicht gleich einfällt, gibst du schnell auf?	<input type="checkbox"/> ja <input type="checkbox"/> nein
Möchtest du programmieren lernen?	<input type="checkbox"/> ja <input type="checkbox"/> nein
Warum möchtest Du programmieren lernen?	

Fragebogen Problemlöseverhalten

Beginnzeit:

Versuche möglichst viele Aufgabe zu lösen. Aus folgenden Gruppen solltest du mindestens 1 Aufgabe lösen:

Gruppe A: 1, 8, 15

Gruppe B: 2, 6,7

Gruppe C: 3,4,5,12

Gruppe D: 10,11,13

Gruppe E: 14

Aufgabe 1

4 Kinder (Anton, Bruno, Christoph und Daniel) wollen im Dunkeln über eine Brücke gehen. Sie haben aber nur 1 Taschenlampe, mit deren Hilfe jeweils maximal 2 Personen sicher über die Brücke gehen können. Zusätzlich haben sie unterschiedliche Gehgeschwindigkeiten: Anton benötigt 1 Minute, Bruno 2 Minuten, Christoph 5 und Daniel 10 Minuten um die Brücke zu überqueren. Wenn 2 Personen gemeinsam gehen, muss sich die Schnellere der Langsameren anpassen. Wie schaffen sie es, dass alle Kinder innerhalb von 17 Minuten die Brücke überqueren?

Aufgabe 2

In einem Raum befinden sich 8 Leute. Bei der Verabschiedung gibt jeder jedem die Hand. Wie oft werden die Hände geschüttelt?

Beispiel 3:

Auszählreim

6 Kinder A(nton), B(erta), C(laudia), D(aniel), E(gon) und F(ritz) stellen sich im Kreis auf, um mit einem 4-silbigen Auszählreim auszuführen, z. B.: 1-2-3-weg

Begonnen wird bei Anton mit 1.

- Welches Kind bleibt als Letztes übrig?
- Welche sind die beiden Letzten?)
- Kann es sein, dass (bei geeigneter Anfangsperson) Anton und Egon die letzten beiden sind?)
- Die beiden Letzten bilden ein Team. Welche Teams können entstehen?

Beispiel 4:

Auf einem Platz mit quadratischen Platten (Schachbrettmuster) veranstalten einige Kinder ein Hüpfspiel. z.B.: 3 Platten vorwärts, Vierteldrehung nach rechts, dann 2 Platten vorwärts, Vierteldrehung nach rechts, 1 Platte vorwärts, Vierteldrehung nach rechts, dann wieder von vorne beginnen. Kurzschreibweise: 3-2-1-3-2-1-3-2-1 usw.

- a) Wo ist das Kind nach 24 Sprüngen?
- b) Was passiert bei der Hüpffolge 4-3-2-1-4-3-2-1-...

Beispiel 5:

Max geht jeden Dienstag spazieren. Moritz geht jeden 5. Tag spazieren (also an 1 Tag spazieren, dann 4 Tage Pause, usw.). Heute sind sie sich begegnet.

- a) Wie lange dauert es, bis sie wieder zusammentreffen können?
- b) Kann es vorkommen, dass sie in einem Monat 2x aufeinandertreffen?
- c) Gibt es Monate, in denen sie nie aufeinandertreffen?
- d) Wie oft können sie sich in einem Jahr begegnen?
- e) Sind Angaben möglich, bei denen sie sich nie oder viel öfter begegnen?

AUFGABE 6:

Daniel hat eine Modelleisenbahn mit 2 verschiedenen Lokomotiven und 5 verschiedenen Waggons. Er stellt immer nur Züge mit 1 Lokomotive und 2 Waggons zusammen. Die Lok kommt immer an den Anfang eines Zuges. Wie viele verschiedene Züge kann er auf diese Art zusammenstellen?

AUFGABE 7:

Im Korb mit den gewaschenen Socken liegen alle lose herum. Insgesamt liegen dort 6 Paar weiße und 12 Paar schwarze Socken. Wie oft musst du höchstens hineingreifen, um ein Paar gleichfarbiger Socken in den Händen zu haben?

AUFGABE 8:

Die Badewanne ist voll Wasser. Mutti gibt mir einen 3-Liter Behälter und einen 5-Liter Behälter. Beide haben keine Maßangaben. Sie sagt: "Bringe mir 4 Liter Wasser".

AUFGABE 9:

Mitten in der großen Pause hat Ute eine SMS von Valentin bekommen. Erika, die Dritte im Bunde, die gerne mehr wissen möchte über deren Handy-Nachrichten versucht bei beiden ihr Glück, aber niemand will ihr verraten, um was es in der SMS ging. Nach langem Bitten und Betteln gibt Ute der neugierigen Erika folgende Auskunft: Ich habe lediglich „GERNE“ zurück geschrieben.

Das macht Erika nur noch neugieriger. Nun versucht Erika bei Valentin ihr Glück. Nachdem sie ihn lange genug nervt, zeigt er ihr die folgenden beiden SMS-

Nachrichten mit dem Kommentar: „Du wirt sicher nicht drauf kommen... wir haben nämlich unsere Geheimsprache benutzt!“

25 08286 154 28978 853 83386		08468
Absender: VALENTIN		Absender: UTE

Aufgabe 15

Drei Forscher werden von Indianern gefangen genommen und an drei hintereinander stehende Marterpfähle gebunden. Dabei sieht der vorderste Forscher keinen der anderen beiden Kollegen. Der zweite sieht einzig den Materpfahl mit dem vordersten Forscher. Der dritte sieht vor sich die beiden Marterpfähle mit seinen beiden Kollegen. Der Mediziner steckt nun jedem der Forscher eine Feder ins Haar, die er versteckt für den jeweiligen Forscher aus seinem Beutel herauszieht.

Nun spricht der Mediziner zu ihnen: "Insgesamt hatte ich zwei schwarze und drei weiße Federn in meinem Beutel. Ich gebe euch die Chance frei zu kommen, wenn einer begründet weiß, welche Farbe die Feder hat, die er angesteckt bekam. Aber wehe ihr könnt die Antwort nicht begründen, dann werdet ihr alle einen qualvolleren Tod als normalerweise finden!" Sehr lange herrscht Stille, doch dann schreit einer: "Hurra, jetzt weiß ich es!"

Welcher Forscher war es und welche Farbe hatte seine Feder?

Aufgabe 10

Welches Ergebnis liefern folgende Anweisungen?

X=1

Solange $x < 5$ wiederhole

Schreibe x

Vergrößere x um 1

Ende Wiederhole

Aufgabe 11

a=12

b=9

wiederhole

wenn $a > b$

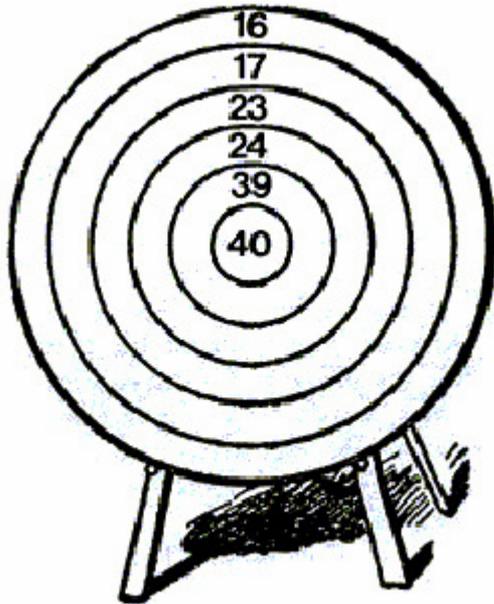
dann verkleinere a um den Wert von b

sonst verkleinere b um den Wert von a

schreibe a, b

bis $a=0$ oder $b=0$

Aufgabe 12

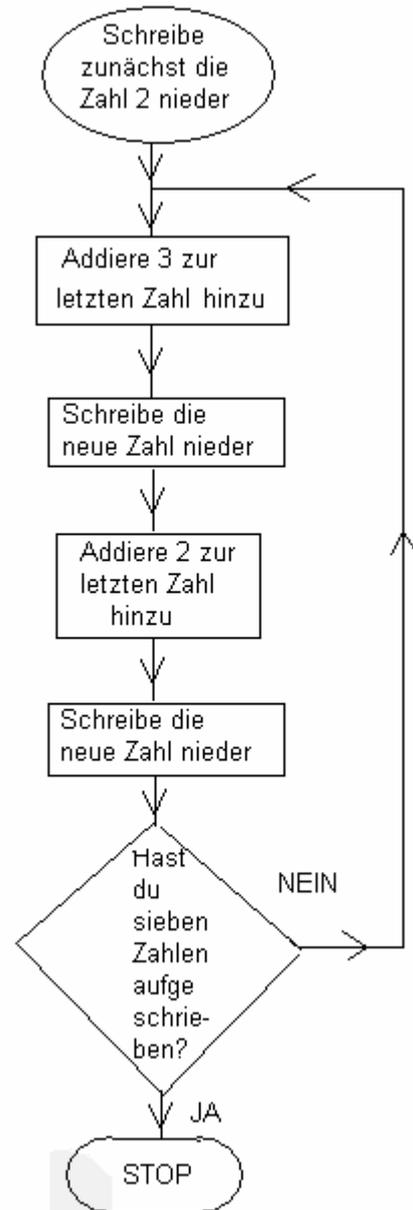


Wie viele Pfeile braucht man um auf der Zielscheibe genau 100 Ringe zu erzielen?

Lösung:

$$16 + 16 + 17 + 17 + 17 + 17 = 100$$

Aufgabe 13



Lösung: 2,5,7,10,12,15,17

Aufgabe 14

Beschreibe einen Vorgang

Verwende dabei Redewendungen wie;

WENN DANN SONST

WIEDERHOLE MAL

WIDERHOLE BIS

1. Zähne Putzen
2. Mit dem Handy telefonieren
3. Eine Datei ausdrucken
4. Ein Auto starten und damit wegfahren

Denke an mögliche Probleme

1. Keine Zahnpasta zu Hause, keine Zahnbürste, .
2. Keine Verbindung, Es meldet sich niemand, ...
3. Kein Papier, Toner/Tinte aus, ...
4. Batterie leer, Benzin aus, ...

Ende Zeit:

<i>Frage</i>	<i>Antwort</i>
Wie viele Aufgaben hast du gelöst	
Wie viele Aufgabe glaubst du richtig gelöst zu haben	
Hast du bei der Lösung intensiv nachgedacht	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hast du die Aufgaben gerne gelöst	<input type="checkbox"/> sehr gerne <input type="checkbox"/> gerne <input type="checkbox"/> nicht gerne <input type="checkbox"/> überhaupt nicht gerne
Welche Aufgaben sind dir leicht gefallen? (Nummer)	
Welche Aufgaben sind dir schwer gefallen? (Nummer)	
Hast Aufgaben nicht gemacht	<input type="checkbox"/> nein, ich habe alle gemacht <input type="checkbox"/> ja, weil sie zu schwer waren <input type="checkbox"/> ja, weil mir nicht gleich eine Lösung eingefallen ist <input type="checkbox"/> ja, weil ich glaubte, dass ich nicht alleine auf die Lösung komme <input type="checkbox"/> ja, weil ich nicht gerne nachdenke <input type="checkbox"/> ja, weil es mir zu viele Aufgaben waren

Beobachtung durch den Lehrer

<i>Frage</i>	<i>Antwort</i>
Name des Schülers	
Eifer	<input type="checkbox"/> sehr eifrig <input type="checkbox"/> eifrig <input type="checkbox"/> nicht eifrig
Ausdauer	<input type="checkbox"/> sehr <input type="checkbox"/> mittel <input type="checkbox"/> nicht ausdauernd
Hat verschiedene Lösungswege versucht	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat sich mit den Aufgaben intensiv auseinander gesetzt	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat bald aufgegeben	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat bald andere Dinge gemacht	<input type="checkbox"/> ja <input type="checkbox"/> nein
War mit Freude dabei	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat verweigert	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat sich geärgert	<input type="checkbox"/> ja <input type="checkbox"/> nein
Sonstige Beobachtungen das Problemlöseverhalten betreffend	

Der Fragebogen nach dem Programmierunterricht

Fragebogen am Ende des Projekts „Programmieren Unterstufe“

Allgemeiner Teil

<i>Frage</i>	<i>Antwort</i>
Geschlecht	<input type="checkbox"/> männlich <input type="checkbox"/> weiblich
Alter	
Klasse	
Hast Du schon einmal ein Computerprogramm geschrieben?	<input type="checkbox"/> ja <input type="checkbox"/> nein
Wenn ja: In welcher Programmiersprache	
Wenn ja: Versuche zu erklären, was Programmieren ist	
Spielst du gerne mit dem Computer	<input type="checkbox"/> ja <input type="checkbox"/> nein
Welche Spiele spielst du?	
Spielst du Schach oder Ähnliches	<input type="checkbox"/> ja <input type="checkbox"/> nein
Löst du gerne Denksportaufgaben	<input type="checkbox"/> sehr gerne <input type="checkbox"/> gerne <input type="checkbox"/> nicht gerne
Löst du neue mathematische Aufgaben?	<input type="checkbox"/> ja <input type="checkbox"/> nein
Kommst du dabei selbst auf Lösungen	<input type="checkbox"/> ja <input type="checkbox"/> nein
Wie lange (Minuten, Stunden) kannst du dich mit einem Problem (Denksportaufgabe, Mathematik, ...) beschäftigen?	
Wie schätzt du dich beim Lösen solcher Probleme ein?	<input type="checkbox"/> sehr gut <input type="checkbox"/> gut <input type="checkbox"/> mittelmäßig <input type="checkbox"/> schlecht
Wenn dir die Lösung nicht gleich einfällt, gibst du schnell auf?	<input type="checkbox"/> ja <input type="checkbox"/> nein
Möchtest du programmieren lernen?	<input type="checkbox"/> ja <input type="checkbox"/> nein
Warum möchtest Du programmieren lernen?	

Fragebogen Problemlöseverhalten

Beginnzeit:

Versuche möglichst viele Aufgabe zu lösen. Aus folgenden Gruppen solltest du mindestens 1 Aufgabe lösen:

Gruppe A: 1, 8, 15

Gruppe B: 2, 6,7

Gruppe C: 3,4,5,12

Gruppe D: 10,11,13

Gruppe E: 14

Aufgabe 1

In einem Raum befinden sich 7 Leute. Bei der Verabschiedung gibt jeder jedem die Hand. Wie oft werden die Hände geschüttelt?

Beispiel 2:

Auszählreim

6 Kinder A(nton), B(erta), C(laudia), D(aniel), E(gon) und F(ritz) stellen sich im Kreis auf, um mit einem 5-silbigen Auszählreim auszuführen, z. B.: 1-2-3-4-weg

Begonnen wird bei Anton mit 1.

- Welches Kind bleibt als Letztes übrig?
- Welche sind die beiden Letzten?
- Kann es sein, dass (bei geeigneter Anfangsperson) Anton und Egon die letzten beiden sind?)

Beispiel 3:

Max geht jeden Dienstag spazieren.

Moritz geht jeden 4. Tag spazieren (also an 1 Tag spazieren, dann 4 Tage Pause, usw.).

Heute sind sie sich begegnet.

- Wie lange dauert es, bis sie wieder zusammentreffen können?
- Kann es vorkommen, dass sie in einem Monat 2x aufeinandertreffen?

Beispiel 4:

Daniel hat eine Modelleisenbahn mit 2 verschiedenen Lokomotiven und 4 verschiedenen Waggons. Er stellt immer nur Züge mit 1 Lokomotive und 2 Waggons zusammen. Die Lok kommt immer an den Anfang eines Zuges. Wie viele verschiedene Züge kann er auf diese Art zusammenstellen?

AUFGABE 5:

Im Korb mit den gewaschenen Socken liegen alle lose herum. Insgesamt liegen dort 7 Paar weiße und 10 Paar schwarze Socken. Wie oft musst du höchstens hineingreifen, um ein Paar gleichfarbiger Socken in den Händen zu haben?

Aufgabe 6

Welches Ergebnis liefern folgende Anweisungen?

X=1

Solange $x < 7$ wiederhole

 Schreibe x

 Vergrößere x um 1

Ende Wiederhole

Aufgabe 7

a=24

b=18

wiederhole

 wenn $a > b$

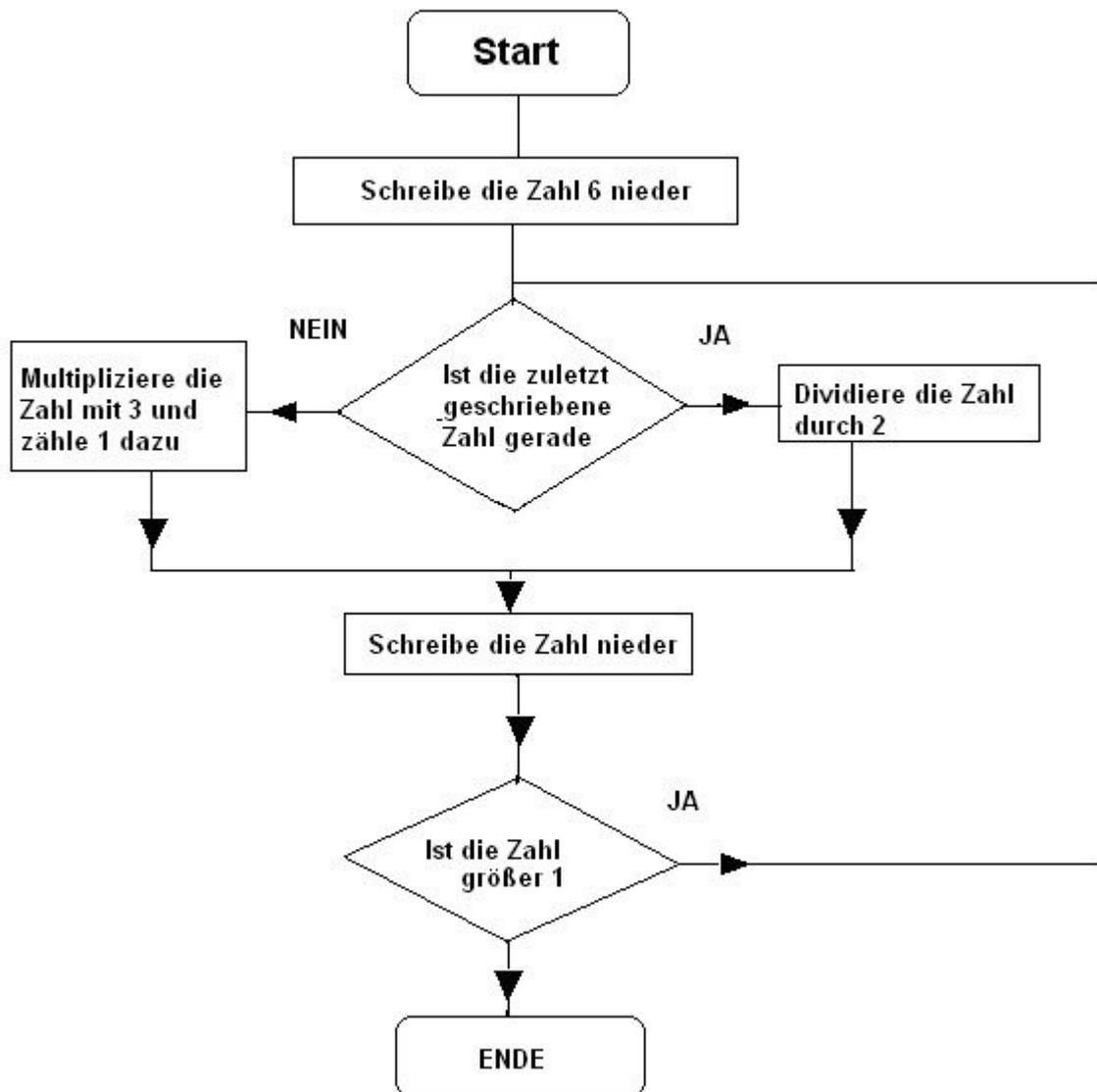
 dann verkleinere a um den Wert von b

 sonst verkleinere b um den Wert von a

 schreibe a, b

bis $a=0$ oder $b=0$

Aufgabe 8



Starte nun nicht mit der Zahl 6 sondern mit der Zahl 7.

Aufgabe 9

Beschreibe **einen** Vorgang, den du beim letzten Test nicht beschrieben hast.

Verwende dabei Redewendungen wie;

WENN DANN SONST

WIEDERHOLE MAL

WIDERHOLE BIS

1. Zähne Putzen
2. Mit dem Handy telefonieren
3. Eine Datei ausdrucken
4. Ein Auto starten und damit wegfahren

Denke an mögliche Probleme

1. Keine Zahnpasta zu Hause, keine Zahnbürste, .
2. Keine Verbindung, Es meldet sich niemand, ...
3. Kein Papier, Toner/Tinte aus, ...
4. Batterie leer, Benzin aus, ...

Ende Zeit:

<i>Frage</i>	<i>Antwort</i>
Wie viele Aufgaben hast du gelöst	
Wie viele Aufgabe glaubst du richtig gelöst zu haben	
Hast du bei der Lösung intensiv nachgedacht	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hast du die Aufgaben gerne gelöst	<input type="checkbox"/> sehr gerne <input type="checkbox"/> gerne <input type="checkbox"/> nicht gerne <input type="checkbox"/> überhaupt nicht gerne
Welche Aufgaben sind dir leicht gefallen? (Nummer)	
Welche Aufgaben sind dir schwer gefallen? (Nummer)	
Hast Du Aufgaben nicht gemacht	<input type="checkbox"/> nein, ich habe alle gemacht <input type="checkbox"/> ja, weil sie zu schwer waren <input type="checkbox"/> ja, weil mir nicht gleich eine Lösung eingefallen ist <input type="checkbox"/> ja, weil ich glaubte, dass ich nicht alleine auf die Lösung komme <input type="checkbox"/> ja, weil ich nicht gerne nachdenke <input type="checkbox"/> ja, weil es mir zu viele Aufgaben waren

Fragen zum Programmierunterricht

<i>Frage</i>	<i>Antwort</i>
Das Programmieren habe ich gerne gemacht	<input type="checkbox"/> sehr gerne <input type="checkbox"/> gerne <input type="checkbox"/> nicht so gerne <input type="checkbox"/> überhaupt nicht
Ich möchte einen Fortsetzungskurs (eventuell mit einer anderen Programmiersprache)	<input type="checkbox"/> sehr gerne <input type="checkbox"/> gerne <input type="checkbox"/> nicht so gerne <input type="checkbox"/> überhaupt nicht
Das Programmieren war für mich	<input type="checkbox"/> sehr leicht <input type="checkbox"/> leicht <input type="checkbox"/> schwer <input type="checkbox"/> sehr schwer
Das Interesse am Programmieren	<input type="checkbox"/> ist größer geworden <input type="checkbox"/> gleich geblieben <input type="checkbox"/> geringer geworden <input type="checkbox"/> habe nun kein Interesse mehr daran

Beobachtung durch den Lehrer

<i>Frage</i>	<i>Antwort</i>
Name des Schülers	
Eifer	<input type="checkbox"/> sehr eifrig <input type="checkbox"/> eifrig <input type="checkbox"/> nicht eifrig
Ausdauer	<input type="checkbox"/> sehr <input type="checkbox"/> mittel <input type="checkbox"/> nicht ausdauernd
Hat verschiedene Lösungswege versucht	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat sich mit den Aufgaben intensiv auseinander gesetzt	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat bald aufgegeben	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat bald andere Dinge gemacht	<input type="checkbox"/> ja <input type="checkbox"/> nein
War mit Freude dabei	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat verweigert	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat sich geärgert	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat sich am Verhalten des Schülers / der Schülerin etwas geändert?	

Fragen zum Unterricht auf den Schüler/die Schülerin bezogen

Frage	Antwort
Eifer	<input type="checkbox"/> sehr eifrig <input type="checkbox"/> eifrig <input type="checkbox"/> nicht eifrig
Ausdauer	<input type="checkbox"/> sehr <input type="checkbox"/> mittel <input type="checkbox"/> nicht ausdauernd
Hat versucht die Aufgaben selbst zu lösen	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat es geschafft, Aufgaben selbst zu lösen	<input type="checkbox"/> ja <input type="checkbox"/> nein
Hat über den Unterricht hinaus Aufgaben gelöst	<input type="checkbox"/> ja <input type="checkbox"/> nein

Fragen an den Lehrer/die Lehrerin

Frage	Antwort
Die Schüler für das Programmierprojekt zu gewinnen war leicht?	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> mit Werbeaufwand
Das Interesse der SchülerInnen	<input type="checkbox"/> nahm ab <input type="checkbox"/> blieb gleich <input type="checkbox"/> wurde größer
Es wäre sinnvoll, Programmieren in der Unterstufe mit diesen Programmen weiterhin anzubieten	<input type="checkbox"/> ja <input type="checkbox"/> nein
Die SchülerInnen würden gerne im nächsten Jahr wieder mitmachen	<input type="checkbox"/> ja <input type="checkbox"/> nein viele einige
	<input type="checkbox"/> ja <input type="checkbox"/> nein