

Bild 1.

Von einer befreundeten Firma erfährt die Firma *ProfiSoft*, dass die neue JAVA-Version 5.0 so genannte *generische Datentypen* erlaubt. Die bisherige Klasse wird folgendermaßen modifiziert: Anstelle eines speziellen Datentyps wie `int`, `float`, `char` oder `Object` setzt sie einen Typ-Stellvertreter – auch *Typparameter* oder *Typvariable* genannt – ein. Der Name des Stellvertreters wird in der Klassendeklaration hinter dem Klassennamen angegeben, eingeschlossen in spitze Klammern. Sie wählt den Buchstaben `T` (für `Typ`) als Stellvertreter und erhält nun folgende *generische Klasse*:

```
class DatenspeicherMinMax <T> {
    private T Min, Max;
    void setMax (T neuerWert)
        {Max = neuerWert;}
    void setMin (T neuerWert)
        {Min = neuerWert;}
    T getMax()
        {return Max;}
    T getMin()
        {return Min;}
} // Ende Datenspeicher
```

Anstelle eines konkreten Typs steht jetzt einfach `T`. Die Deklaration des Typnamens steht nur einmal hinter dem Klassennamen. Damit sie ihre speziellen Klassen erhält, erzeugt sie sich nun diese Klassen aus der generischen Klasse (siehe Bild 1).

Hinter dem Klassennamen wird – wie bei der Klassendeklaration – in spitzen Klammern der konkrete Typ angegeben. Alle generischen Eigenschaften besitzt jetzt der angegebene Typ. Erfreut stellt die Firma *ProfiSoft* fest, dass die oben aufgeführten zwei Probleme damit verschwunden sind: Die Typsicherheit ist wieder vorhanden, da nur Werte mit dem angegebenen Typ in `Min` und `Max` gespeichert werden können – keine gemischten. Außerdem entfällt beim Lesen die explizite Typanpassung. Allerdings können generische Typen nur Referenztypen sein; es ist also nicht möglich, `<int>` oder `<float>` zu schreiben. Solche generischen Klassen werden als *einfache Klassenschablonen* bezeichnet. Der Name des Typ-Stellvertreters muss in der Klassendeklaration angegeben werden, da es mehr als einen Stellvertreter geben kann!

```
DatenspeicherMinMax <Float> einFloatDatenspeicherMinMax =
    new DatenspeicherMinMax <Float>();
DatenspeicherMinMax <Integer> einIntegerDatenspeicherMinMax =
    new DatenspeicherMinMax <Integer>();
DatenspeicherMinMax <Character> einCharacterDatenspeicherMinMax =
    new DatenspeicherMinMax <Character>();
```

Das Problem, dass bei generischen Klassen keine einfachen Typen verwendet werden können, wird in JAVA 5.0 durch das so genannte *Auto Boxing* gelöst. Einfache Werte werden von JAVA 5.0 selbstständig in Objekte umgewandelt und umgekehrt.

Bedeutung generischer Typen für die Ausbildung

Wie die Beispiele gezeigt haben, sind generische Typen ein Konzept zur Verallgemeinerung von Programmen. Man programmiert sozusagen auf einer höheren Abstraktionsebene. Für Schüler und Schülerinnen bedeutet dies, dass sie lernen, ihre Programme zu verallgemeinern. Im zweiten Schritt sind sie dann in der Lage, Programme von vornherein generisch zu konzipieren.

In JAVA 5.0 werden in der JCF (Java Collection Framework) generische Typen intensiv verwendet. Dieses „Rahmenwerk“ stellt eine einheitliche Architektur zur Repräsentation und Manipulation von Sammlungen (*collections*) zur Verfügung. Ohne Kenntnis des generischen Konzepts kann man die JFC in Zukunft nicht geeignet einsetzen. Daher kommt man – auch in der Schule – ohne generische Typen nicht aus.

Helmut Balzert

Literatur und Internetquellen

Balzert, H.: Lehrbuch Grundlagen der Informatik. Heidelberg u. a.: Elsevier, 2004.

Balzert, H.: JAVA 5 – Der Einstieg in die Programmierung. Herdecke; Bochum: W3L-Verlag, 2005.

JAVA Specification Request 14 (JSR 14) [hier ist die Spracherweiterung definiert]: JSR 14: Add Generic Types To The JAVA™ Programming Language: <http://jcp.org/en/jsr/detail?id=14> [Stand: November 2004]

Geschichte

Der Apfel war vergiftet

Aus Leben und Werk von Alan Mathison Turing

Zu den größten Mathematikern, die England je hervor gebracht hat, zählt Alan Turing. Zugleich gilt er als einer der wegweisendsten Computerpioniere. Doch das Leben Turings verlief in keiner Weise geradlinig.

Außenseiter

Der Vater Alan Turings, Julius Mathison Turing, ein britischer Staatsbeamter in der damaligen Kronkolonie Indien, und seine Frau Ethel beschlossen, dass ihr Kind unbedingt in England geboren wer-

Alan M. Turing (1912–1954).



<http://antzia.com/History/Biography/Turing/turing.jpg>

den sollte, reisten deshalb nach London, und am 23. Juni 1912 wurde Alan Mathison Turing dort geboren. Ein halbes Jahr nach seiner Geburt kehrten seine Eltern wieder nach Indien zurück und überließen ihn und seinen fünfjährigen Bruder John der Obhut eines pensionierten Colonels und dessen Frau. Der junge Alan kam mit dieser Situation jedoch nicht zurecht. Er begann zu stottern und entwickelte sich zu einem exzentrischen Eigenbrötler, der nicht in der Lage war, sich an die damaligen gesellschaftlichen Gepflogenheiten in England anzupassen.

Seine besonderen Begabungen zeigten sich allerdings schon früh. Obwohl Alan beim Lesen und Schreiben Schwierigkeiten hatte, brachte er sich diese Fähigkeiten mit starker Selbstdisziplin selbst bei. Bereits mit sieben Jahren konnte er alle sechsstelligen Seriennummern der Straßenlaternen in seinem Wohnort auswendig. Mit neun Jahren gelang es ihm, die Höhen von Hügeln und Bergen fast exakt zu berechnen. Mit elf Jahren begeisterte er sich für organische Chemie und destillierte beispielsweise aus Tangpflanzen, die er sich aus dem Meer holte, Jod. Andererseits musste sich Alan einen Punkt auf den Daumen malen, damit er links von rechts unterscheiden konnte.

Nach mehreren Schulwechslern besuchte er ab seinem 14. Lebensjahr die *Sherborne School*, eine Schule, die darauf Wert legte, die künftige Elite des britischen Empires heranzuziehen. Turing wurde dort als schüchtern und ungeschickter Junge mit Talent auf naturwissenschaftlichem Gebiet beurteilt. Seine Begabungen und Interessen wurden allerdings von den Lehrkräften nicht gefördert, und der Rektor der *Sherborne School* äußerte über ihn (vgl. O'Connor/Robertson, 2003): „Wenn er ein reiner Wissenschaftler werden will, dann vergeudet er seine Zeit in einer staatlichen Schule.“

Turing-Maschine

Trotz seiner nicht konfliktfreien Schuljahre bekam Turing im Jahr 1931 einen Studienplatz am *King's College* in Cambridge, wo er sich

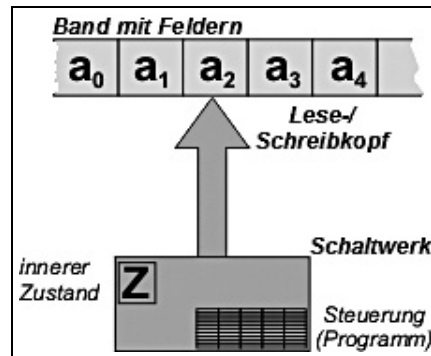
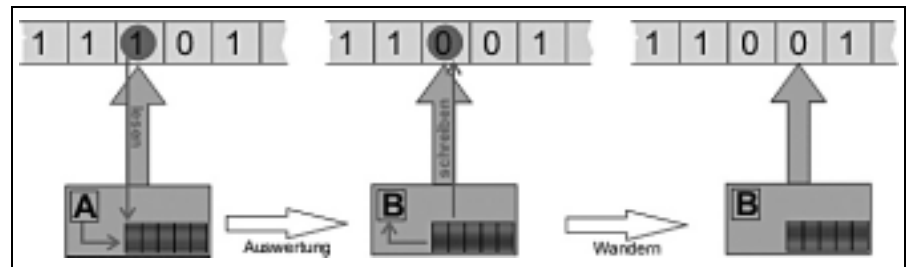


Bild 2:
Die Turing-Maschine (links) und ihre grundsätzliche Funktionsweise (unten).

nach:
http://www.mathe.tu-freiberg.de/~dempe/schuelerpr_neu/turing.htm



vor allem mit den Grundlagen der Mathematik und Logik auseinandersetzte. Dabei stand die *Frage der Unentscheidbarkeit* im Mittelpunkt. Diskutiert wurde zu jener Zeit, dass die Mathematik nicht die allmächtige Wissenschaft war, für die sie gehalten wurde, da es auch nicht entscheidbare und damit unlösbare Probleme geben könnte.

In Zusammenhang mit diesen Auseinandersetzungen schrieb Turing 1936 seinen einflussreichsten mathematischen Aufsatz „On Computable Numbers, with an application to the Entscheidungsproblem“ („Über berechenbare Zahlen mit einer Anwendung auf das Entscheidungsproblem“), der 1937 veröffentlicht wurde (vgl. Turing, 1936/1937). Er unternahm darin den Versuch, die unentscheidbaren Fragen ausfindig zu machen, indem er eine imaginäre Maschine beschrieb. Diese Maschine konnte eine bestimmte Operation ausführen (z. B. addieren), indem in diese Maschine zunächst über einen Papierstreifen Zahlen eingelesen werden sollten und ein Ergebnis wiederum über den Papierstreifen ausgegeben werden konnte. Darauf aufbauend konzipierte Turing eine Maschine, die ihre interne Arbeitsweise ändern konnte – es war die von ihm so genannte *universal computing machine* entstanden, die später nach ihm benannte *Turing-Maschine*. Diese Maschine ist in der Lage,

grundsätzlich jede Frage zu beantworten, die logisch zu beantworten ist.

Im Prinzip besteht eine Turing-Maschine (siehe Bild 2) aus

- ▷ einem unendlich langen Speicherband mit unendlich vielen, nebeneinander angeordneten Feldern. In jedem dieser Felder kann genau ein Zeichen gespeichert werden; das Feld kann aber auch leer sein.
- ▷ einem Schaltwerk, das eine endliche Zahl von Zuständen einnehmen kann. Es steuert das Verhalten der Turing-Maschine.
- ▷ einem Lese- und Schreibkopf, der auf dem unendlich langen Speicherband ein Feld nach links oder rechts rücken, das in einem Feld stehende Zeichen lesen, in ein Feld hineinschreiben, das Zeichen löschen oder es stehen lassen kann.

Zur Steuerung ihrer Aktionen – d. h. zum Bewegen und Schreiben, Lesen usw. – besitzt die Turing-Maschine eine endliche Menge von Zuständen und definierten Zustandsübergängen. Die Zustände und ihre Zustandsübergänge sind in einer Zustandsänderungstabelle enthalten, die als gespeicherter Algorithmus bzw. als gespeichertes Programm angesehen werden kann. In Abhängigkeit von dem Zustand, in dem sich das Schaltwerk bzw. die

Steuereinheit befindet, und dem gelesenen Zeichen auf dem Speicherband wird eine Aktion ausgeführt, und die Turing-Maschine geht in einen neuen Zustand über (siehe Bild 2, vorige Seite).

Turing beschrieb in seinem Aufsatz, wie die Maschine für Entscheidungsprobleme eingesetzt werden könne, das heißt für Fragen, die ausschließlich mit „Ja“ oder „Nein“ zu beantworten sind. Dabei wird das Anhalten der Turing-Maschine als „Ja“ interpretiert und das „Nein“ entsprechend durch das Nicht-Anhalten, d.h. durch eine Endlosschleife. Jedes Problem lässt sich somit als Entscheidungsproblem formulieren, indem gefragt wird, ob ein bestimmter Wert eine Lösung für ein konkretes Problem ist. Insoweit stellt jede Turing-Maschine einen speziellen Algorithmus dar. Der US-amerikanische Mathematiker Alonzo Church (1903–1995) stellte darüber hinaus die These auf, dass ein Algorithmus genau dann berechenbar ist, wenn die zugehörige Turing-Maschine nach endlich vielen Schritten anhält. Mit anderen Worten: Gilt diese These, so sind alle berechenbaren Funktionen genau diejenigen Funktionen, die durch Turing-Maschinen berechnet werden. Mit dieser so genannte *Church'schen These*, die eigentlich Turing-Church-These heißen müsste, ist es jedoch nicht möglich, jedes unentscheidbare Problem zu finden – genau die anfangs gestellte Frage bleibt also offen. Bewiesen ist diese Hypothese zwar bis heute nicht, sie wird jedoch grundsätzlich als gültig angenommen. Trotzdem lieferte Turing durch seine Gedankengänge sozusagen die „Blaupause“ des modernen programmierbaren Computers (vgl. u.a. Lehmann, 1999) – er war zu jener Zeit erst 24 Jahre alt.

Turing beendete 1936 seine Studien am *King's College* und erhielt für seine Arbeiten den *Smith's Prize* der Cambridge University.

Zur Promotion ging Turing von 1938 bis 1939 nach Princeton (USA) und lernte dort neben Alonzo Church auch John von Neumann (1903–1957) persönlich kennen, der das Potenzial der Arbeiten Turings sofort erkannte und ihn ermutigte, sein theoretische Modell mechanisch umsetzen zu lassen.

Turing-Bomben und COLOSSUS

Nach dem Ausbruch des zweiten Weltkriegs wurde Turing als Leiter der Abteilung für Kryptoanalyse für das Projekt „Ultra“ in *Bletchley Park*, einem unter dem Decknamen „Station X“ geheim gehaltenen Ort nördlich von London, berufen. Mit „Ultra“ wurden vom englischen Geheimdienst *MI6* alle Vorgänge bezeichnet, die mit dem Abhören und Decodieren von Funkprüchen der Kriegsgegner – vor allem der Deutschen, aber auch der Italiener und Japaner – zu tun hatten. In Bletchley Park arbeiteten Linguisten, Mathematiker, Verschlüsselungsspezialisten, ja sogar bekannte Schachspieler, um sich mit der Entschlüsselung der feindlichen Funkprüche zu befassen. Anfangs wurden dort ungefähr 200 Personen beschäftigt, gegen Ende des Krieges waren es ca. 7000.

Klar ist mittlerweile geworden, dass die europäische Geschichte einen anderen Verlauf genommen hätte, wenn nicht Turing an diesem Projekt so maßgeblich beteiligt gewesen wäre – vielleicht wäre sogar die erste Atombombe nicht auf Japan, sondern auf Deutschland abgeworfen worden.

Vor allem in Deutschland wurde eine Maschine zur Verschlüsselung eingesetzt, die einen nahezu mystischen Ruf genoss: die *ENIGMA* (griechisch für „Geheimnis“ – vgl. Batzer, 1996; Witten/Letzner/Schulz, 1999; Schneider/Koerber, 2000; Wikipedia, 2004). Die *ENIGMA* bestand aus einer Schreibmaschinentastatur und mehreren Walzen, die elektrische Kontakte besaßen. Wurde eine Taste gedrückt, so floss Strom von der Taste durch die Walzen bis zu einer Anzeige, wo ein Buchstabe aufleuchtete (siehe Bild 3). Die angezeigten Buchstaben bildeten den ver- bzw. entschlüsselten Text. Da sich bei jedem Tastendruck die Walzen weiterdrehten, wurde der gleiche Buchstabe immer wieder anders verschlüsselt. Zunächst wurden mit der *ENIGMA* Nachrichten mithilfe von drei drehbaren Walzen verschlüsselt; später wurden von der deutschen Marine bis zu acht Walzen benutzt. Über 100000 Geräte dieser Art wurden in Deutschland während des Krieges eingesetzt.

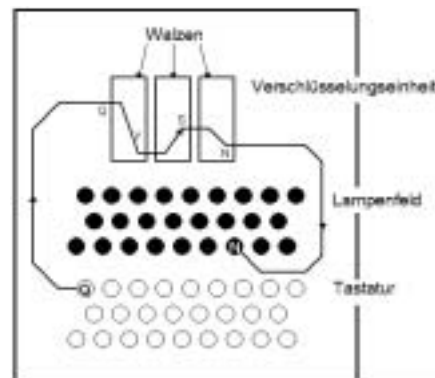


Bild 3: Funktionsweise der ENIGMA. Später wurden bis zu acht Walzen eingesetzt.

Die *ENIGMA*, eine Entwicklung des deutschen Ingenieurs und Unternehmers Arthur Scherbius (1878–1929), die zunächst als zivil-kommerzielles Chiffriersystem – z.B. für Banken – gedacht war, wurde bereits ab 1925/26 von der deutschen Wehrmacht, aber auch von Regierungsstellen als Standardverschlüsselungsmaschine eingesetzt. Und schon vor dem zweiten Weltkrieg hatte der polnische Versicherungsmathematiker Marian Rejewski (1905–1980) das Geheimnis im Grundsatz entschlüsselt. Dazu setzte er eine Reihe von ihm entwickelter mechanischer Rechenmaschinen ein, die von ihm „bomby“ (deutsch: Bomben) genannt wurden und die einen Teil der täglichen Rechenarbeit übernahmen. Diese „Bomben“ lieferten innerhalb von zwei Stunden den Tagesschlüssel, der zum Verschlüsseln von Nachrichten diente und von den Deutschen täglich um 0 Uhr gewechselt wurde. Aus Angst vor Invasionen war nämlich Anfang der Zwanzigerjahre in Polen das *Biuro Szyfrów* gegründet worden, dessen Aufgabe es war, den deutschen und russischen Funkverkehr zu überwachen. Für diese Stelle arbeitete Rejewski, der bei Ausbruch des zweiten Weltkriegs nach England flüchtete und sein Wissen sofort in Bletchley Park weitergab. Zur Kryptoanalyse wurde deshalb die dann auf Englisch „bomby“ genannten und nunmehr als *Turing-Bomben* bekannten mechanischen Dechiffriermaschinen entwickelt und eingesetzt (siehe Bild 4, nächste Seite).

http://www-ivs.cs.uni-magdeburg.de/bs/lehre/wise0102/prcgb/vortraege/jain/Bilder/turing_bombe.jpg

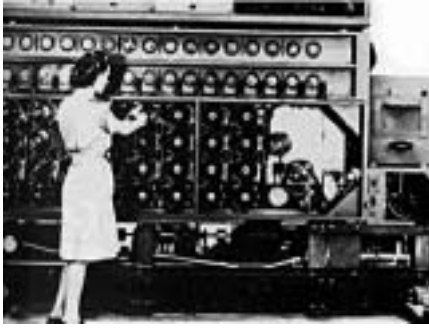


Bild 4: Die so genannten Turing-Bomben.

Die Turing-Bomben waren der polnischen Entwicklung sogar überlegen, weil sie nicht mehr nur mechanisch, sondern bereits mit Relais arbeiteten. Allerdings hatten sie den Nachteil, dass es, wenn sie gestoppt wurden, keine hundertprozentige Wahrscheinlichkeit mehr gab, dass die richtigen Walzenpositionen zur Entschlüsselung der ENIGMA-Zeichenfolge erkannt werden konnten. Trotzdem war die Decodierung so erfolgreich, dass etliche falsche Fährten vom englischen Geheimdienst gelegt werden mussten, damit die deutsche Kriegsführung nicht den Verdacht schöpfen konnte, die ENIGMA sei entschlüsselt.

Zur geheimen Kommunikation der obersten deutschen Heeresleitung, d. h. zwischen Hitler und seinen Generälen wurde allerdings

Bild 5: Die Lorenz SZ40 – eine Chiffriermaschine, die im zweiten Weltkrieg von der obersten Heeresleitung Deutschlands eingesetzt wurde.

http://www.alanturing.net/turing_archive/graphics/LorenzMachine.jpg



nicht die ENIGMA, sondern ein Fernschreiber mit der Bezeichnung *Lorenz SZ40* benutzt (siehe Bild 5). Nach ersten Analysen der abgefangenen verschlüsselten Informationen stellte Turing statistische Theorien auf, mit denen bereits ein kleiner Teil entschlüsselt werden konnte. Doch mit herkömmlichen mechanischen oder elektromechanischen Maschinen war es nicht möglich, in vertretbaren Zeiträumen den Code zu bearbeiten. So wurde beschlossen, auf elektronischem Weg mit magnetischen Speichern zu arbeiten. Anhand von Turings Konzept einer universellen Maschine – der Turing-Maschine – entwarf der ebenfalls in Bletchley Park arbeitende Mathematiker Max Newman (1897–1984) eine ausschließlich elektronisch gesteuerte Maschine, die *COLOSSUS* (später *COLOSSUS I* genannt).

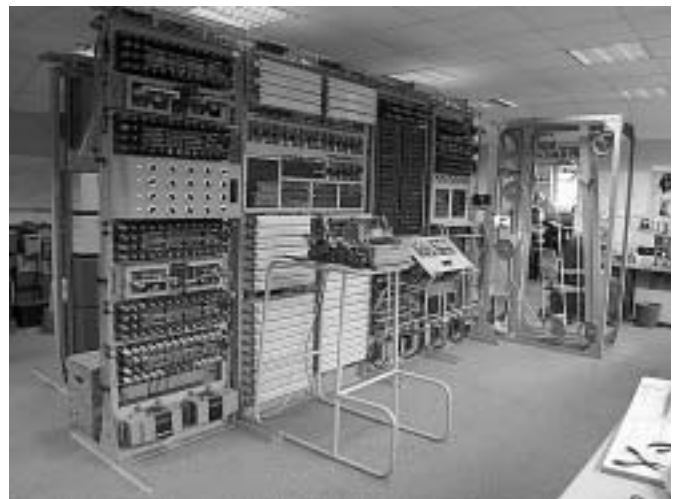
Gebaut wurde dieser Computer im Forschungszentrum der britischen Post und 1943 in Bletchley Park in Betrieb genommen. *COLOSSUS I* hatte 1500 Röhren, das Folgemodell *COLOSSUS II* 2500 Stück (siehe Bild 6). Zwischen 1943 und 1946 wurden insgesamt zehn Maschinen gebaut. *COLOSSUS* erlaubte die Entschlüsselung einer Nachricht innerhalb weniger Stunden: Im Allgemeinen lieferte die Maschine in nur zwei Stunden die konkreten Einstellungen der Lorenz SZ40. Es konnten 5000 Zeichen pro Sekunde von der *COLOSSUS* verarbeitet werden, wobei jedes Zeichen aus 5 Bit bestand und photoelektrisch von einem Lochstreifen gelesen wurde. Damit war die *COLOS-*

SUS wesentlich schneller als die von Relaisaltern abhängigen Turing-Bomben. Die *COLOSSUS* gilt zwar als der erste speicherprogrammierbare Computer, war jedoch fest an eine bestimmte Aufgabe angepasst und nicht im heutigen Sinn frei programmierbar. Das erste Gerät wurde unter strengen Geheimhaltungsrichtlinien gebaut, sodass keinerlei Aufzeichnungen oder Handbücher darüber mehr existieren. Alle Maschinen wurden 1946 aus Geheimhaltungsgründen demontiert und vernichtet. Erst 1970 wurde überhaupt bekannt, dass es *COLOSSUS* gab.

Während die erste *COLOSSUS* zwar von den Ideen Turings inspiriert war, ist er an der Entwicklung dieser Maschine zunächst nicht beteiligt gewesen. Am 1. Juni 1944 wurde eine neue Maschine, die *COLOSSUS II*, in Betrieb genommen. Sie wurde von Turing konstruiert. Die Baupläne wurden von der britischen Regierung erst 55 Jahre nach dem Ende des zweiten Weltkriegs, im Jahr 2000, freigegeben. Ein Nachbau der *COLOSSUS II* ist nunmehr im Museum des Bletchley Parks zu sehen (siehe Bild 6).

Die Entschlüsselung des Funkverkehrs der Deutschen hatte entscheidenden Einfluss auf den Verlauf des zweiten Weltkriegs. So war die britische Luftabwehr stets im Voraus über die deutschen Angriffspläne im Luftkrieg um England informiert, und 1940 war die zunächst überlegene deutsche Luftwaffe so geschwächt, dass sie jegliche Angriffe einstellte. Ähnliches geschah im Atlantik: Die Positionen

Bild 6: Mit dem ersten Röhrenrechner COLOSSUS – hier der Nachbau der von Turing entworfenen COLOSSUS II im Museum von Bletchley Park – konnte die Codierung der Lorenz SZ40 entschlüsselt werden.



<http://www.qedata.se/bilder/antiklar/londonsviten/lonsv-blet-colossus.jpg>

deutscher U-Boote konnten bestimmt werden, sodass einerseits britische Schiffe sie umfahren und andererseits gezielte Angriffe auf sie gestartet werden konnten – die deutschen Verluste an Schiffen verzehnfachten sich zwischen 1940 und 1944. Ebenso konnten die Nachschubwege des Afrika-Korps des Generals Erwin Rommel (1891–1944) gestört werden, sodass im Mai 1943 die deutschen Streitkräfte in Afrika letztlich kapitulieren mussten.

Für seine Arbeiten in Bletchley Park wurde Turing 1946 mit dem *Order of the British Empire* (OBE) geehrt. Trotzdem wurden Turings Arbeiten an der Entschlüsselung der ENIGMA und der Lorenz bis in die 70er-Jahre des vorigen Jahrhunderts geheim gehalten, und nicht einmal seine engsten Freunde wussten davon.

Turing-Test

Nach dem Krieg arbeitete Turing weiter daran, seine Ideen über einen universellen Computer umzusetzen. Er entschied sich, einen Lehrauftrag der Universität in Cambridge nicht anzunehmen und stattdessen ab 1945 bei einer Arbeitsgruppe mitzuarbeiten, die am *National Physical Laboratory* (NPL) an Entwurf, Konstruktion und Anwendung solcher elektronischen Maschinen arbeitete und sich vor allem aus ehemaligen Mitgliedern des COLOSSUS-Teams zusammensetzte. In den drei Jahren seiner Anstellung bis 1948 entwarf Turing den ersten Plan der *Automatic Computing Engine* (ACE), einer Computer-Entwicklung des NPL, und leistete darüber hinaus Pionierarbeit beim Entwurf der Software.

Mit Beginn des Jahres 1949 wurde Turing stellvertretender Direktor der Computerabteilung der *Universität von Manchester*, wo er zunächst vor allem an der Software für einen der ersten echten Computer arbeitete: die *Manchester Automatic Digital Machine* – MADAM. Seine Prognose war, dass diese Maschine spätestens im Jahr 2000 das Gedächtnis eines Menschen nachbilden könne und auch selbst zu denken imstande wäre, denn wäh-



Bild 7: Struktur des Turing-Tests – der Experte stellt eine Aufgabe oder führt einen Dialog und muss entscheiden, ob er einen Menschen oder einen Computer als Kommunikationspartner hatte.

nach:
http://www.alanturing.net/turing_archive/graphics/turingtest.gif

rend dieser Zeit widmete er sich auch weiterhin theoretischen Arbeiten. In „Computing machinery and intelligence“ („Rechenmaschinen und Intelligenz“) griff Turing die Problematik der künstlichen Intelligenz auf und schlug den später nach ihm benannten *Turing-Test* als Kriterium dafür vor (vgl. Turing, 1950): Ein Computer sei dann intelligent, wenn er Aufgaben so bewältigt, dass ein Mensch dies als intelligente Leistung bewerte. Computer verhalten sich jedoch nur in Abhängigkeit ihrer Programmierung. Turings Schlussfolgerung war deshalb, dass – unter der Prämisse, Menschen verhielten sich selbst intelligent – ein Computerprogramm dann als intelligent zu bezeichnen sei, wenn es sich wie ein Mensch verhält, also sein Verhalten von dem eines Menschen nicht unterscheidbar sei. Der Test selbst ist deshalb sehr einfach durchzuführen: Ein Experte muss entscheiden, ob eine bestimmte Leistung von einem Computer oder einem Menschen erbracht wurde (siehe Bild 7). Kann er dies nicht, so besitzt der Computer die gleiche geistige Fähigkeit wie der Mensch.

Bislang hat allerdings noch kein Computerprogramm den Turing-Test vollständig bestanden.

Der Science-Fiction-Autor Philip K. Dick (1928–1982) griff übrigens 1968 eine Variante des Turing-Tests in seinem Roman „Träumen Roboter von elektrischen Schafen“ auf, aus dem 1982 der Film „Blade Runner“ entstand. In einer fernen Zukunft werden dort künstliche Menschen, die physisch den Menschen gleichen, einem Empathie-Test unterzogen, der durch lange Befragungen ihre emotionale Reaktion prüft und hervorbringen soll, ob sie Mensch oder Android sind.

Im Jahr 1951 wurde Turing aufgrund seiner Leistungen Mitglied der *Royal Society*.

Während der Arbeit mit dem Computer MADAM entwarf Turing eine Fülle an Programmen, die intelligentes Verhalten zeigen sollten. So sollte die Maschine u. a. Liebesbriefe entwerfen – hier ein kurzes Beispiel: „Du bist mein gierig begeistertes Gefühl der Gemeinsamkeit. Meine Zuneigung hält sich neugierig an deinen leidenschaftlichen Wunsch. Mein Gefallen sehnt sich nach deinem Herzen. [...]“

Allerdings entwarf Turing 1952 auch erste Schachprogramme, um die „Intelligenz“ von Computern an der Spielstärke im Schach messen zu können. Doch Turings Schachprogramm – das erste vollständige Schachprogramm der Welt – konnte von MADAM und anderen damals schon existierenden Computern nicht abgearbeitet werden. Turing simulierte deshalb selbst in einer – noch erhalten gebliebenen – Partie den Computer und errechnete die Züge auf dem Papier anhand seiner Programmierung. Es ist überliefert, dass er ungefähr eine halbe Stunde für die Berechnung jedes einzelnen Zugs benötigte, den die Maschine aufgrund seines Programms ausführen würde. Die „Papier-Maschine“ verlor diese historische Partie gegen einen Kollegen Turings (die Partie ist mit Kommentaren über den LOG-IN-Service, siehe Seite 128, zu erhalten.)

Tragischer Tod

Dass Turing kein einfacher Mensch war, zeigte sich bei allen seinen Wirkungsstätten. So gab es in Bletchley Park immer wieder

<http://www.univie.ac.at/bv/photo-gallery/jpeg/44%20Turing,%20Weitlaur.JPG>



Bild 8: Turing 1949 beim Marathon-Lauf.

Streitigkeiten mit dem Militär, da er Menschen gänzlich ignorierte, von deren Intellekt er nichts hielt – und Militärangehörige zählten für ihn zu dieser Kategorie. Und sie meinten, er nähme seine Aufgabe nicht ernst, da er immer so aussah, als hätte er im Freien übernachtet: zerzaustes Haar, verdreckte Hände und eine mit einem alten Schulschlips hochgebundene Hose (er fuhr mit dem Fahrrad zur Arbeit). Allerdings war er körperlich sehr fit. So wäre er beinahe für die Olympischen Spiele in Helsinki 1952 nominiert worden, weil er in einem Qualifikationslauf 1949 die Marathon-Strecke in 2 h 46 min 03 s lief, das heißt in einer Zeit, die nur 10 Minuten über derjenigen lag, die der Marathon-Sieger zu den Spielen 1948 in London benötigt hatte.

Turings Karriere wurde jäh vernichtet, nachdem er 1952 der Polizei den Einbruch in sein Haus gemeldet hatte. Während der Ermittlungen stellte sich zwar heraus, dass ein Freund Turings einem Komplizen die Tipps zum Einbruch gegeben hatte, aber die Polizei machte Turing den Vorwurf von „grober Unzucht und sexueller Perversion“ und unterstellte ihm homosexuelle Beziehungen zu diesem Freund. Turing verteidigte sich nicht und wurde verurteilt, entweder ins Gefängnis zu gehen oder sich einer Hormonbehandlung zu unterziehen. Die Medikamente, für die er sich

entschied und die ihn von seiner „Krankheit“ heilen sollten, veränderten seinen Körper völlig. Trotz seiner Kompetenz wurde er auch von jeglicher Arbeit mit Computern ausgeschlossen.

Kurz vor seinem 42. Geburtstag starb Alan Mathison Turing am 7. Juni 1954 in seinem Haus in Wilmslow an einer Cyanid-Vergiftung. Ein halb aufgeessener Apfel, der mit dem Cyanid präpariert war, lag neben ihm. Die polizeilichen Untersuchungen wurden mit dem Ergebnis „Selbstmord“ eingestellt.

Viele Werke Turings sind erst später nach seinem Tod oder auch noch gar nicht publiziert worden. Eine dreibändige Kollektion „Collected Works of A. M. Turing“ – „Mechanical Intelligence“, „Morphogenesis“ und „Pure Mathematics“ – erschien 1992.

Zum Gedächtnis

Während Turing zu seinen Lebzeiten zwar einige Ehrungen erfuhr, konnten doch seine Verdienste erst in vollem Umfang gewürdigt werden, als vor allem seine Arbeiten in Bletchley Park nach dem Aufheben der Informationssperre in den 70er-Jahren bekannt wurden. So wurde an seinem 50. Todestag, dem 7. Juni 2004, zum Gedenken eine Tafel an seinem früheren Haus „Hollymeade“ in Wilmslow enthüllt. Und bereits am 23. Juni 2001 wurde eine Turing-Statue in Manchester aufgestellt.

Darüber hinaus wird jährlich bereits seit 1966 der Turing-Preis von der *Association for Computing Machinery* (acm) an Personen verliehen, die bedeutende Arbeiten im Informatikbereich geleistet haben. Dieser offiziell als *A. M. Turing Award* bezeichnete und mit 100 000 Dollar dotierte Preis wird weithin als Nobelpreis für die Welt der Computer angesehen. Der erste Preisträger war Alan J. Perlis (geb. 1922) für fortgeschrittene Programmier- und Compiler-Techniken und für Compilerbau.

Bernhard Koerber

Im LOG-IN-Service (siehe S. 128) finden Sie eine Wiedergabe der ersten Computer-Schachpartie mit Kommentaren.

Literatur und Internetquellen

Batzer, P.: Die ENIGMA – Grundlagen zu einer Unterrichtssequenz. In: LOG IN, 16. Jg. (1996), S. 44–51.

Bletchley Park – Homepage:
<http://www.bletchleypark.org.uk/page.cfm?pageid=159>
[Stand: Dezember 2004]

Lehmann, E.: Die Turing-Maschine im Anfangsunterricht – Ein Bericht von den ersten Stunden eines Informatikkurses in Klasse 11. In: LOG IN, 19. Jg. (1999), H. 6, S. 44–52.

O'Connor, J. J.; Robertson, E. F.: Alan Mathison Turing.
<http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Turing.html>
[Stand: Dezember 2004]

Schneider, M.; Koerber, B.: Streng geheim – Die Welt der verschlüsselten Kommunikation. In: LOG IN, 20. Jg. (2000), H. 2, S. 4–5.

Schulz, A.: Die goldene Gans, die niemals schnattert.
<http://www.chessbase.de/nachrichten.asp?newsid=3245>
[Stand: Dezember 2004]

Simulator der ENIGMA:
http://www.wisspro.de/iug_site/themen/sicherheit/beitraege/ENIGMAApplet.htm
[Stand: Dezember 2004]

Simulator der Turing-Maschine:
<http://wwwsys.informatik.fh-wiesbaden.de/weber1/turing/tm.html>
[Stand: Dezember 2004]

The Alan Turing Home Page:
<http://www.turing.org.uk/turing/>
[Stand: Dezember 2004]

Turing, A. M.: On computable numbers, with an application to the Entscheidungsproblem. In: Proceedings of the London Mathematical Society, 42. Jg. (1936/1937), Serie 2, S. 230–265.
<http://www.abelard.org/turpap2/tp2-ie.asp>
[Stand: Dezember 2004]

Turing, A. M.: Computing machinery and intelligence. In: Mind, 59. Jg. (1950), Nr. 236, S. 443–460.
<http://www.abelard.org/turpap/turpap.htm>
[Stand: Dezember 2004]

Wikipedia – Alan Turing:
http://de.wikipedia.org/wiki/Alan_Turing
[Stand: Dezember 2004]

Wikipedia – ENIGMA (Maschine):
http://de.wikipedia.org/wiki/ENIGMA_%28Maschine%29
[Stand: Dezember 2004]

Witten, H.; Letzner, I.; Schulz, R.-H.: RSA & Co. in der Schule – Moderne Kryptologie, alte Mathematik, raffinierte Protokolle (Teil 3: Flußchiffren, perfekte Sicherheit und Zufall per Computer). In: LOG IN, 19. Jg. (1999), H. 2, S. 50–57.