

Informatik in der Sekundarstufe 2 am Max-Planck-Gymnasium

Die Schülerinnen und Schüler werden zur kritischen und verantwortungsvollen Analyse, Modellierung und Implementierung komplexer Informatiksysteme befähigt. Dabei geht es auch um die Auseinandersetzung mit Fragen einer menschengerechten Gestaltung und der Sicherheit von Systemen. Schülerinnen und Schüler sollen zukünftige Entwicklungen nutzen, verstehen und beurteilen lernen und sich aktiv an der Fortentwicklung beteiligen.

Der Informatikunterricht ist durch eine Handlungs- und Produktorientierung gekennzeichnet, da sich die Schülerinnen und Schüler mit Problemstellungen auseinandersetzen und diese unter Anwendung von fachspezifischen Methoden wie der objektorientierten Modellierung und Programmierung in JAVA systematisch lösen. So wird die Problemlösefähigkeit der Schülerinnen und Schüler verbessert. Es werden reale Problemstellungen aus der Lebenswelt der Schülerinnen und Schüler aufgegriffen. Zudem wird das selbstständige Arbeiten und insbesondere im Rahmen von Projekten die Team-, Kommunikations- sowie die Kooperationsfähigkeit gefördert. Um eine reflektierte Auseinandersetzung mit dem Fach Informatik zu gewährleisten, werden die Schülerinnen und Schüler dazu angeregt, ihre Problemlösungen und bestehende Informatiksysteme zu analysieren und zu bewerten. Dabei werden die Grenzen von Informatiksystemen und auch der verantwortungsvolle Umgang mit ihnen thematisiert.

JAVA ist seit einigen Jahren bereits die für das Abitur in NRW vorgeschriebene Programmiersprache. In einzelnen Unterrichtsprojekten können aber gelegentlich auch andere Programmiersprachen Anwendung finden.

JAVA gehört zu den beliebtesten Programmiersprachen im Internet, weil es einerseits durch seine konsequente Objektorientierung sehr leistungsfähig ist, andererseits viele Eigenschaften hat, die es im Internet einfach und universell einsetzbar machen. Beispielsweise ist es plattformunabhängig und der Compiler kostenlos. Am häufigsten kommt JAVA in Applets vor.

In der Einführungsphase (EF) steht das Erlernen der wesentlichen Konzepte der Programmiersprache JAVA im Vordergrund. Dabei wird zunächst die Lernbibliothek Greenfoot verwendet, die einen direkten spielerischen Einstieg in objektorientierte Denkweise erlaubt. Im zweiten Halbjahr wird dann auf dieses Hilfsmittel verzichtet und direkt in einem normalen JAVA-Editor programmiert. Am Ende des Jahres kann der Programmierkurs als abgeschlossen bezeichnet werden. In den Folgejahren finden die Programmierkenntnisse und –fähigkeiten weiterhin Anwendung und Vertiefung in Projekten zu ausgewählten Themen der Informatik, wie Netzwerkprogrammierung, dynamische Datenstrukturen, Grenzen der Automatisierung, Vergleich von Sortierverfahren, etc.

Für den Informatikunterricht werden keine Vorkenntnisse aus der Sekundarstufe 1 benötigt. Eine wesentliche Voraussetzung ist vor allem die Freude am angewandten Problemlösen und die Freude am Programmieren. Über die drei Jahre lernen die Schülerinnen und Schüler, die Logik der Programmierung zu verstehen und diese Kenntnisse und Fähigkeiten so anzuwenden, dass der Computer tut, was er soll. Informatik ist ein Fach, in dem sehr viel diskutiert wird. Die wichtigste Grundlage dabei ist, die Probleme (fachsprachlich korrekt und) präzise darzustellen. Wie man bei einem kurzen Blick in einschlägige Foren im Internet leicht feststellen kann, haben Informatiker eine sehr eigene Sprache. Vor ihr die Scheu zu verlieren und sich in dieser Welt sicher bewegen zu können, ist ein weiteres Ziel von Informatikunterricht.

In beinahe jeder Unterrichtsstunde wird im Informatikunterricht programmiert.

Bewertungskriterien kompetenzorientiert – Beispiel für das Ende der Einführungsphase (Jahrgang 10)

Note „gut“

Argumentieren: Der Schüler kann die Notwendigkeit spezifischer Werkzeuge oder Methoden **begründen**, wenn diese bereits behandelt worden sind; auf Nachfrage kann er Vor- und Nachteile der verwendeten Methoden (in einer Implementierung) gegenüber Alternativen benennen.

Modellieren: Er kann neue Probleme **häufig selbstständig** in Teilprobleme und daraus in Teilaufgaben zerlegen (Spezifikation und Modularisierung); siehe auch „Implementieren“.

Implementieren: Er kann Methoden und Werkzeuge bereits bekannter Problemlösungen **ohne Hilfestellung** auf ähnliche Aufgabenstellungen anwenden; die Java-Syntax, die Klassenmethoden der bekannten Klassen und die Parameter der bereits verwendeten Methoden sind ihm so geläufig, dass häufig die Compilermeldungen ausreichen um ihn seine Fehler erkennen und selbstständig lösen zu lassen; er löst gestellte (Programmier-) Aufgaben selbstständig, zielorientiert und **weitestgehend** angemessen; sein Programmierstil ist strukturiert und sein Code enthält insbesondere an „schwierigen“ Stellen verständliche Kommentare; Klassen, Variable, Objekte und Methoden erhalten sinnvolle Namen.

Darstellen und Interpretieren: Er verfügt über ein **fundiertes** Verständnis der angesprochenen Probleme; auftauchende Verständnisprobleme ist er bemüht, **sofort** zu beheben und kann sie **nachvollziehbar und eindeutig** benennen; die Veranschaulichung der Probleme unterstützt er **gelegentlich** durch geeignete Grafiken und Diagramme.

Kommunizieren und Kooperieren: Er kann der Diskussion um Lösungswege **nahezu ausnahmslos** folgen; er beherrscht aktiv wie passiv **weitestgehend** die angemessene Terminologie; wo ihm die passenden Termini unbekannt sind, greift er auf die Alltagssprache zurück und bleibt doch insbesondere in der Formulierung von Algorithmen und Regeln **immer eindeutig**; die Beherrschung der Fachsprache zeigt er in mündlichen Stundenwiederholungen und –Zusammenfassungen, Stundenprotokollen, in Tests, Hausaufgaben und seiner Bearbeitung von Klausuraufgaben; er beteiligt sich **mehrheitlich aktiv** an allen Phasen und Arbeitsformen des Unterrichts und **berät schwächere Mitschüler**; er beteiligt sich zielgerichtet, kooperativ und produktiv an der Arbeit in Projektgruppen.

Note „ausreichend“

Argumentieren: Der Schüler kann seine speziellen Lösungswege teilweise schlüssig begründen.

Modellieren: Er kann **neue Probleme** und deren Lösung, ihre Zerlegung in Teilprobleme (Spezifikation und Modularisierung) sowie die Notwendigkeit des Einsatzes spezifischer Werkzeuge oder Methoden in der Regel nur **nachvollziehen**; siehe auch „Implementieren“:

Implementieren: Programmieraufgaben löst er in der Regel, indem er sich eng **an Vorgaben und bewährte Klassen und Methoden** hält. Er kann einen **wesentlichen** Teil der bekannten Methoden und Werkzeuge (z.B. gegen Ende der EF, Eingabe und Ausgabe mit InputDialog und MatDialog etc.) **ohne Hilfestellung** auf neue Aufgabenstellungen **anwenden**, benötigt aber bei der spontanen Lösung deutlich veränderter Fragestellungen im Regelfall immer und auch bei nur geringfügig veränderter Aufgabenstellung häufig konkrete Hilfe. Die Java-Syntax ist ihm dennoch so geläufig, dass ihm vor allem kleinere Fehler meist spätestens mit Hilfe der Compilermeldung auffallen und er sie in der Regel **ohne fremde Hilfe** beheben kann. Vereinzelt seltenere Fehler behebt er **selbstständig**, muss aber **in der Regel die Hilfe** von Mitschülern, Nachschlagewerken oder der Lehrkraft hinzuziehen.

Kommunizieren und Kooperieren: Er kann neue Probleme und deren Lösung, ihre Zerlegung in Teilprobleme (Spezifikation und Modularisierung) sowie die Notwendigkeit des Einsatzes spezifischer Werkzeuge oder Methoden in der Regel nur **nachvollziehen**.

Er kann einen **wesentlichen** Teil der verwendeten Methoden und Objekte in ihrer **Funktion erläutern** und deren Einsatz begründen. Er kann auftretende **Probleme nachvollziehbar beschreiben** und damit die Diskussion im Plenum **bereichern**.

Er **beteiligt sich** an der Arbeit in Projektgruppen und ist dabei bemüht, die Gruppe nicht zu stören.

Darstellen und Interpretieren:

Das **grundlegende** Verständnis der angesprochenen Probleme ist **teilweise lückenhaft**. Der Schüler ist **meist** in der Lage eine schriftlich fixierte Lösung nachvollziehbar zu erläutern und kann diese **häufig** durch geeignete Grafiken und Diagramme unterstützen.